

85
1978

MTA Számítástechnikai és Automatizálási Kutató Intézet Budapest





MAGYAR TUDOMÁNYOS AKADEÉMIA
SZÁMITÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI KUTATÓ INTÉZETE

SZÁMITÓGÉPES SZERSZÁMGÉPVEZÉRLÉS EGY
ALKATRÉSZPROGRAMOZÁSI MÓDSZERE

Irta:

DR HOFFMANN PÉTER

Tanulmányok 85/1978.

A kiadásért felelős:
DR VÁMOS TIBOR

ISBN 963 311 072 6
ISSN 0324-2951

Köszönetek

Köszönetem fejezem ki Nemes Lászlónak és dr. Hatvany Józsefnek, akik szakmai fejlődésemet kezdettől fogva irányították és segítették, akiknek bátorítása nélkül ez a dolgozat nem született volna meg.

Dr. Horváth Mátyás több éve figyelemmel kíséri ezt a munkát, kritikai megjegyzései, segítsége a munka bázisa, motorja.

Munkatársaimmal folytatott beszélgetések, viták a munkát katalizálták.

Időrendi sorrendben Biró Gyula, Mandler György, Bródy Ferenc, Krammer Gergely, Vaskovics György, Farkas Ernő, Mátyási Gyula nevét szeretném említeni.

Szakmai képzésemért kiváló tanárainknak tartozom köszönettel:

V.B.Szmolov /Leningrádi Elektrotechnikai Intézet, H.B. Voelcker,

J.A. Feldmann /Rochesteri Egyetem/.

T A R T A L O M J E G Y Z É K

	Oldal
1. Célkitűzés	7
2. A numerikusan vezérelt forgácsolási folyamat	9
2.1 Bevezetés	9
2.2 Munkadarab, szerszám, szerszámgép	10
2.2.1 Merev testek	10
2.2.2 A munkadarab	12
2.2.3 A szerszám	12
2.2.4 Alakváltóztatás forgácsolással	14
2.2.5 A szerszámgép	16
2.3 Esztergálás	17
2.4 Marás	22
2.5 A megmunkálás vezérlése	24
2.6 A megmunkálás pontossága	26
2.7 Technológiai tervezés, alkatrészprogramozás	28
3. A numerikus szerszámgépvezérlő rendszerek és alkat- részprogramozási módszereik fejlődése	31
3.1 Az NC rendszerek rendszertechikája, szolgáltatás- rendszere	32
3.1.1 A fejlődés főbb állomásai	32
3.1.2 A csoportos vezérlések	33
3.1.3 Az egyedi vezérlések	35
3.1.4 CNC berendezések jellegzetességei	37
3.1.5 A csoportos vezérlések fejlődése	39
3.1.6 A számítógépes szerszámgépvezérlések kutatása Magyarországon	41
3.2 Az NC berendezések vezérlési nyelvei	43
3.2.1 Mozgás_kontúrok programozása	43
3.2.2 Szerszámkorrekció	44
3.2.3 Sebességkompenzáció	46

3.2.4	Megmunkálási hibák korrigálása	46
3.2.5	Mozgás csoportok programozása	48
3.2.6	Feltételtől való függés programozása	49
3.2.7	Kapcsolási funkciók	50
3.3	A számítógépes szerszám gépvezérlés programozásának követelményrendszere	50
3.4	Néhány vezérlőprogram formátum általános kritikája	53
4.	A számítógépes szerszám gépvezérlés egy új vezérlési nyelve: a MID nyelv	56
F1.	Függelék: A MID - szerszám gép vezérlési nyelv - részletes specifikációja	64
F1.1	A MID file vertikális strukturája	64
F1.2	A MID nyelv horizontális strukturája	66
F1.2.1	Jelölések	67
F1.2.2	A MID file fejrésze	69
F1.2.3	A MID file törzse	73
F1.2.3.1	Rekordok azonosítása	73
F1.2.3.2	Mozgási utasítások megadása	74
F1.2.3.3	Mozgási rekordokban használatos adatazonosítók jelentése	74
F1.2.3.4	Kapcsolási funkciók programozása	79
F1.2.3.5	Kapcsolási rekordok adatazonosítóinak jelentése	79
F1.2.3.6	A MID file törzsében előforduló egyéb rekordok	84
F1.2.3.7	Mozgás és kapcsolat paraméteres programozása	87
F1.2.3.8	MID változónak értékadás paraméteres kifejezéssel	87
F1.2.4	Alkatrészprogramok alprogramjainak kezelése	88
F1.2.4.1	Szubrutinok definiálása	88
F1.2.4.2	Szubrutinok hívása	89
F1.2.4.3	Műveletelem típusu szubrutinok	90

F2. Függelék: Technológiai típusu rekordok formális	
leírása	92
F2.1 A módszer	92
F2.2 Jelölések	93
F2.3 Szerszámgép és vezérlés állapotvektor komponensei	94
F2.4 Rekordstruktúra	96
F2.5 Egyéb transzformációs szabályok	97
F2.6 Szöveges megjegyzések a transzformációs szabá- lyokhoz	99
F3. Függelék: Műveletelem típusu technológiai alprogramok	
programozási nyelve	101
F4. Függelék: Programozási példák	105
F4.1 Paraméteres mozgásprogramozás MID programban	105
F4.2 Műveletelem típusu szubrutin programozása: furási művelet	106
F4.3 Műveletelem típusu szubrutin programozása: furóciklus és alkalmazása	108
F4.4 Műveletelem típusu szubrutin programozása: kvadratikus B-spline interpolációja	110
Irodalom	112

1. Célkitűzés

A kutatás feladatát az alábbiakban lehet összefoglalni:

A/ Ki kell dolgozni azokat a követelményeket, amelyek a korszerű számítógépes szerszámgépvezérléssel szemben az alkatrészprogramozás oldaláról reálisan támaszthatók.

A követelményeket úgy kell kialakítani, hogy azok az alábbi előnyöket implikálják:

- a vezérlő programoknak a gyártás változó körülményeihez való adaptálhatósága révén javuljon az előállítható alkatrészek minősége, a gyártáshoz szükséges idő pedig a hagyományos /nem számítógépes/ NC egységekhez viszonyítva ne nőjön illetve lehetőleg csökkenjen;
- javuljon a gépcsoport gépeinek kihasználtsága a kissorozatu gyártás körülményei között;
- csökkenjen a gyártáselőkészítés idő- és költségigénye.

Ennek érdekében:

a/ alakuljon ki optimális munkamegosztás az automatikus alkatrészprogramozási rendszer és a vezérlés között úgy, hogy csökkenjen az alkatrészprogramok teljes feldolgozási igénye a magasszintű nyelvtől a vezérlés hardware eszközéig;

b/ a vezérlő programok strukturája segítse elő azok könnyebb szerkeszthetőségét, belőhetőségét;

- az alkatrész programozás módszere segítse elő a jobb ember-gép kapcsolat megteremtését, ezáltal javítsa a gépeket kiszolgáló személyzet munkakörülményeit, e munka felhasználásának hatékonyságát.

B/ Ki kell dolgozni és egy konkrét vezérlő nyelvben realizálni kell a számítógépes szerszámgépvezérlés egy olyan programozási módszerét, amely különösen alkalmas csoportos és egyedi számítógépes szerszámgépvezérlő berendezésekben való alkalmazásra. A nyelvet úgy kell kialakítani, hogy a benne leírt programok /a vezérlő progra-

mok/ közvetlenül képezhessék a számítógépes vezérlés inputját. Minden további feldolgozásnak a vezérlésen belül kell megtörténnie.

Ki kell dolgozni a vezérlési nyelv vertikális és horizontális logikai strukturáját és fizikai reprezentációját. Meg kell határozni, hogy a számítógépes vezérlésben az alkatrészprogram milyen konszekutív feldolgozási fázisokon megy keresztül, és mindegyik fázishoz meg kell határozni az eredménystruktúrát a megfelelő fizikai reprezentációval együtt.

A vezérlési nyelv specifikálásához absztrakt, egyértelmű módszert kell alkalmazni, amely egy automatizált alkatrészprogramozási rendszer számára a vezérlő rendszert és nyelvét definiálja.

A kutatási célkitűzés a számítógépes szerszámgépvezérlő rendszer kifejlesztésének lényeges részfeladata. Ezen keresztül kapcsolódik az 1012/1972. sz. Kormányhatározattal kiemelt K-6: GÉPGYÁRTÁSTECHNOLÓGIA KUTATÁSA FEJLESZTÉSE c. országos szintű kutatási célprogram 3. /forgácsolás/ és 9. /automatizálás/ témacsoportjához.

A célkitűzés konkrét hozzájárulás a célprogram azon célkitűzéseéhez, miszerint javuljon a géppark kihasználása, csökkenjen a gyártáselőkészítés idő- és költségigénye, megteremtődjenek a korszerű üzemszervezési és termelésirányítási rendszerek alkalmazásához szükséges feltételek és a szellemi rutinmunka automatizálása révén elősegítsék az alkotó jellegű technológusi tevékenységet.

2. A numerikusan vezérelt forgácsolási folyamat

2.1 Bevezetés

A fejezet célja a numerikusan vezérelt forgácsolási folyamat modelljének megalkotása és a vezérlési nyelv bevezetése e modell segítségével. A folyamatot az alkatrészprogramozás oldaláról vizsgáljuk, a modellben ezért csak azok a jelenségek, tulajdonságok tükröződnek, amelyek programozhatók vagy programmal befolyásolhatók.

Az alkatrészprogramozás szempontjából irreveláns részletektől tehát elvonatkoztatunk. Az elhagyható ismeretek mennyisége, másszóval a modell mélysége, függ a szerszámgepek és vezérléseik pillanatnyi fejlettségi szintjétől. Ezt a hátrányt ellensúlyozza az a tény, hogy a fejlődéssel általában a modell egyszerűsödése várható: a tendencia ugyanis az, hogy az újabb és újabb vezérlések egyre több funkciót automatizálnak.

E dolgozat tématerületét a numerikusan vezérelt forgácsolás általános területén belül azzal korlátozzuk, hogy csak a szakaszmenti és a gépgyártásban előforduló egyszerűbb pályamenti esztergálás és marás szempontjait vesszük figyelembe. Ebből a tárgyalásból több más forgácsolási eljárásra /pl. a furásra/ vonatkozó modell egyszerűen származtatható, de az összes numerikusan vezérelt megmunkálást nem lehet ezzel a modellel lefedni. A korlátozást azonban a feltárható terület nagysága és jelentősége elfogadhatóvá teszi.

Mindjárt előljáróban meg kell jegyezni, hogy a kitűzött célt még az említett megkötések mellett sem fogjuk tudni teljesen elérni. Az ok egyszerűen az, hogy ilyen modell ma még nem létezik, annak ellenére, hogy kidolgozásra érdekében világszerte már több helyen tettek erőfeszítéseket. Amit el tudunk érni az csak annyi, hogy az ismert eredményeket e dolgozat szempontjai szerint adaptáljuk és továbbfejlesztjük, ilymódon igyekszünk tárgyalásunkhoz egységes, konzisztens és teljes elméleti háttérrel biztosítani. Több helyen csak a probléma felvetéséig tudunk eljutni, a téma nehézsége miatt azonban feltehetően ez is hasznos hozzájárulás lesz a további munkákhoz.

Gondolatmenetünk lényege: forgácsolást merev testek mozgásával magyarázható jelenségnek tekintjük és írjuk le. A forgácsolt munkadarab

alakjának a testek nem merev voltából és egyéb tényezőkből eredő eltéréseit visszavezetjük a mozgás pályájának torzulására. A tényleges pályát egy elvárt névleges pálya és egy "hiba-pálya" szuperpozíciójának tekintjük.

2.2 Munkadarab, szerszám, szerszámgép

2.2.1 Merev testek

A forgácsolási modellben alapvető szerepet játszik a merev testek fogalma.

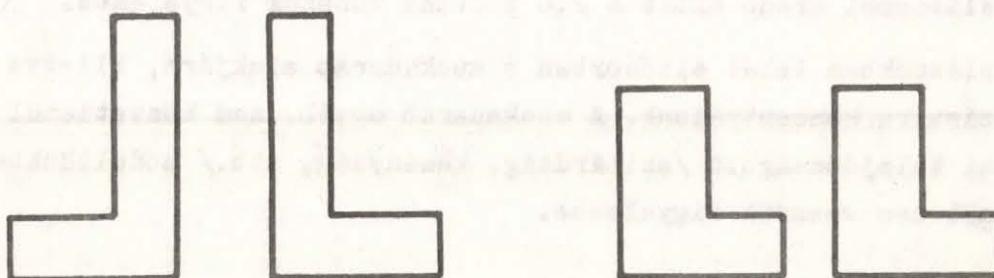
A merev testeket a 3 dimenziós euklideszi tér pontthalmazaival modellezzük. Ezentul maguk a testek helyett mindig a modellező pontthalmazt fogjuk vizsgálni és akkor is a modellező pontthalmazra gondolunk, ha az egyszerűség kedvéért a merev testet említjük.

Ez a modellezés azon a feltételezésen alapszik, hogy egyrészt minden merev testre meghatározható a modellező pontthalmaz és viszont /2.1. ábra/, másrészt a pontthalmazok tulajdonságai leképezései a testek valószínű tulajdonságainak. Ezzel a leképezéssel, illetve annak megalapozottságával ehelyütt nem kívánunk foglalkozni.



2.1. ábra: Merev testek modellezése
pontthalmazokkal

A merev test jellemzője alakja. Két merev test alakja akkor ekvivalens, ha pontjaik között létezik olyan egy-egy értelmű távolságtartó megfeleltetés, amelyben az egymásnak megfelelő lokális koordinátarendszerek irányítása megőrződik. Az alak ezen definíciója független a tárgy térbeli helyzetétől, továbbá kizárja két olyan test alakjának ekvivalenciáját, amelyek a 3D térben történő elmozgatással egymással nem hozhatók fedésbe /2.2. ábra/.



a. Nem ekvivalens
alaku testek

b. Ekvivalens alaku
testek

2.2. ábra: Testek alakjának ekvivalenciája a 2D euklideszi térben

Azokat a /derékszögű/ koordináta-rendszereket, amelyekben a test pontjainak koordinátái állandóak, a merev test saját koordináta-rendszereinek nevezzük.

A merev testek pontthalmazára a továbbiakban a test térbeli helyzetének függvényében lesz szükségünk. Olyan α alakfüggvényt kívánunk használni, amely a test modellező pontthalmazának elemeit a test bármely h térbeli helyzetének függvényében megadja:

$$\alpha : \text{térbeli helyzet} \rightarrow \text{pontthalmaz}$$

Az alakfüggvény számára a térbeli helyzetet a tárgy valamely referenciapontjának térbeli helyzetével és e pont körüli előfordulásainak gömbi koordinátákkal kifejezett mértékével tekintjük adottnak.

2.2.2. A munkadarab

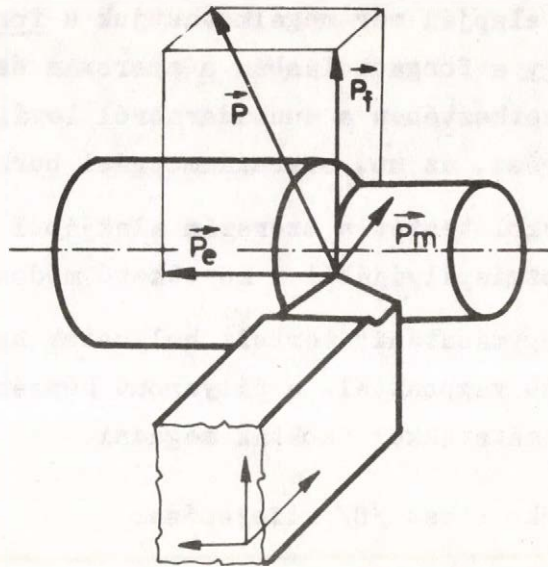
A munkadarab a fenti definíció szerinti merev test, amelyet kiindulási állapotában előgyártmánynak, a megmunkálás utáni állapotában pedig készdarabnak nevezünk. Az előgyártmány és a készdarab alakja különbözik, éppen ezen alakváltoztatás a forgácsolás célja. A munkadarab merevségét állító feltételezés természetesen csak elhanyagolással igaz. A közelítésből eredő hibát a 2.6 pontban vesszük figyelembe.

Tárgyalásunkban tehát elsősorban a munkadarab alakjára, illetve alakváltozásaira koncentrálunk. A munkadarab egyéb, nem közvetlenül geometriai tulajdonságait /szilárdság, keménység, stb./ modellünkben közvetlenül nem vesszük figyelembe.

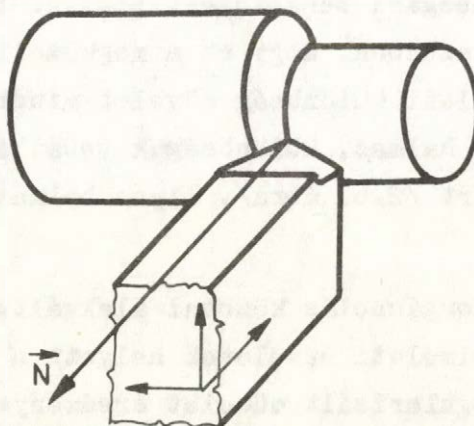
2.2.3. A szerszám

A megmunkálási folyamat alapvetően a munkadarab és egy speciális merev test, a szerszám relatív elmozdulása következtében valósul meg. A szerszám alakja olyan, hogy a munkadarabhoz képest megfelelő relatív helyzetbe hozva megfelelő irányu erő hatására képes a munkadarab anyagába behatolni és arról anyagot leválasztani /2.3. ábra/. A leválasztott anyag a forgács, maga a művelet a forgácsolás. A forgácsolás eredménye pedig az, hogy a munkadarab alakja megváltozik. A szerszámnak ezek szerint alakján kívül fontos jellemzője a forgácsolás megvalósulásához szükséges erő iránya, és a munkadarab megmunkálás alatt álló felületének, illetve e felületnek a megmunkálási pontbani normálisának a szerszámmal képesti relatív helyzete /2.4. ábra/.

A szerszám alakját és egyéb jellemzőit a technológiai tervezés veszi figyelembe, amikor a szerszámtervezést, a szerszámelrendezést stb. tervezi. A szerszám technológiai tervezés szempontjából teljes modelljének létrehozása a tervezés automatizálásának feltétele, amely ma még nem tekinthető megoldottnak.



2.3. ábra: Forgácsolást kiváltó erő komponensei
az esztergakés koordinátarendszerében



Munkadarab és szerszám forgácsolásra alkalmas
relatív helyzete esztergálásnál /a munkadarab
felület N normálisa a megmunkálási pontban/

2.4. ábra: Esztergálás szerszáma

2.2.4. Alakváltoztatás forgácsolással

Az eddig elmondottak alapján már megalkothatjuk a forgácsolás modelljét. Azt mondjuk, hogy a forgácsolásban a szerszám és a munkadarab relatív elmozdulása következtében a munkadarabról leválnak a szerszámmozgás által bejárt térrész, az un. szerszámmozgási burkolótest pontjai.

A szerszámmozgási burkolótestet a szerszám alakjából és a szerszám - munkadarab relatív mozgáspályájából a következő módon származtathatjuk.

A mozgás P pályája "egymásutáni" térbeli helyzetek halmaza. A pályát rendszerint a kezdő és végponttal, a pályapont képzés szabályával és a szabálytól függő paraméterekkel szokták megadni.

A szerszámmozgási burkolótest B kifejezése:

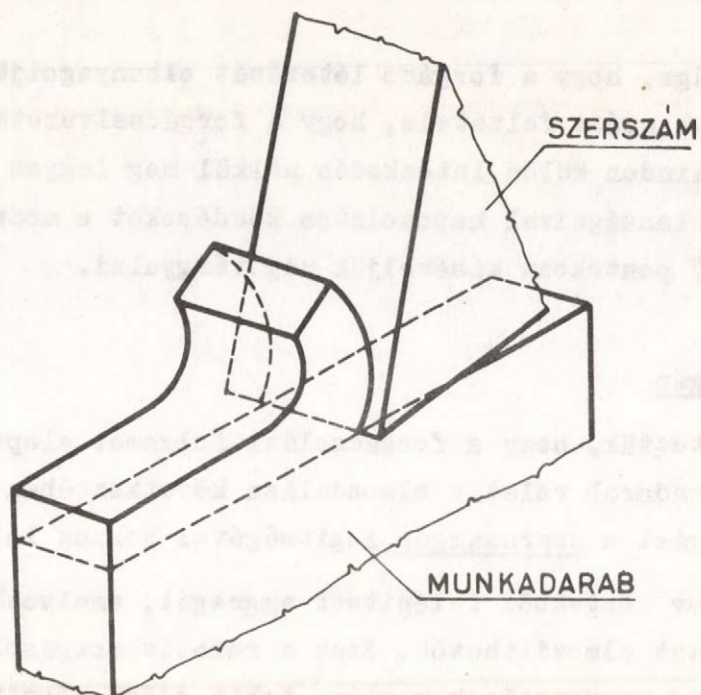
$$B = \bigcup_i \alpha /h_i/, \quad h_i \in P$$

ahol α a 2.2.1.-ben bevezetett alakfüggvény.

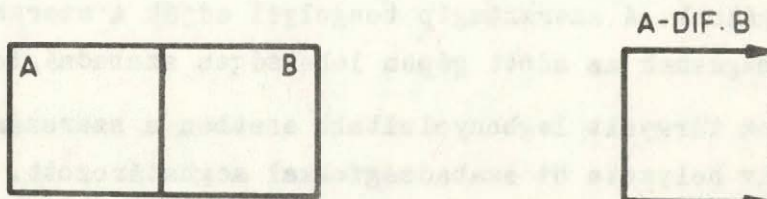
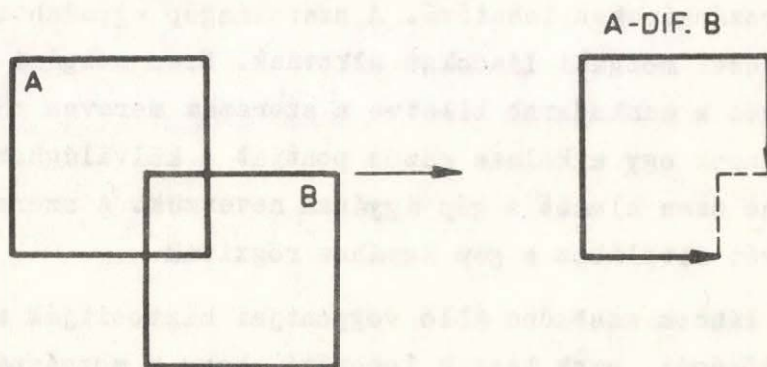
Első pillantásra úgy tűnik, mintha az anyageltávolítási folyamatot matematikailag két ponthalmaz halmazelméleti különbségével lehetne leírni. A különbség művelet kisebbbitendője a munkadarab ponthalmaza, a kivonandó pedig a szerszámmozgási burkolótest pontjai lehetnének /2.5. ábra/. Gyorsan belátható azonban, hogy ez a megközelítés helytelen eredményt ad. A halmazelméleti különbség művelet mindkét operandusa ugyanis topológiailag zárt halmaz, különbségük tehát általános esetben sem nem nyílt, sem nem zárt /2.6. ábra/. Ilyen halmaz pedig nem modellez merev testet.

Voelcker a merev testek forgácsolás közbeni alakváltozásainak leírására a "közönséges" halmazelméleti műveletek helyett a regularizált műveleteket használja. A regularizált művelet eredménye mindenképpen reguláris halmaz: vagyis olyan, amelynek belső pontjai által alkotott halmaz lezártja maga a kiindulási halmaz. Ez a halmazelméleti művelet nem triviális, de jelenleg a probléma egyetlen létező korrekt megoldása.

Forgácsolási modellünkben az alakváltozást egy mozgás eredményeként irtuk le. A módszer előnye, hogy a numerikus vezérlésekben közvetlenül éppen ezt a mozgást kell megadni, programozni.



2.5. ábra: Alakváltoztatás forgácsolással



2.6. ábra: Halmazelméleti különbségképzés lehetséges eredményei zárt operandushalmazok esetében

A modell hiányossága, hogy a forgács létezését elhanyagolja. A modell alkalmazhatóságának ezért feltétele, hogy a forgácselvezetés a technológiai rendszerben minden külön intézkedés nélkül meg legyen oldva. A megmunkálás pontatlanságaival kapcsolatos kérdéseket e modell segítségével a 2.6 és 2.7 pontokban kíséreljük meg tárgyalni.

2.2.5 A szerszámgép

Fentebb már kifejtettük, hogy a forgácsolási folyamat alapvetően a szerszám és a munkadarab relatív elmozdulása következtében valósul meg. Ezt a relatív mozgást a szerszámgép segítségével hozzuk létre.

A szerszámgép merev testekből felépített aggregát, amelynek alkotóelemei egymáshoz képest elmozdithatók. Ezek a relatív mozgások konstrukciós okokból mindig megvezetések mentén, tehát előre rögzített pályán történnek. A leggyakrabban /és az általunk vizsgált területen kizárólagosan/ egyenesek menti megvezetéseket és relatív elfordulásokat tesznek a szerszámgépeken lehetővé. A szerszámgép egymáshoz képest elmozduló alkatrészei mozgási láncokat alkotnak. Ezen mozgási láncok egy-egy végpontjához a munkadarab illetve a szerszám mereven rögzíthető. A mozgási láncok egy alkalmas közös pontját a külvilághoz mereven rögzítik. A lánc ezen elemét a gép ágyának nevezzük. A szerszámgép koordinátarendszerét általában a gép ágyához rögzítik.

A mozgási láncok szabadon álló végpontjai biztosítják a szerszámgép vezérelhetőségét, ezek teszik lehetővé, hogy a mozgáspályákat a mindenkori megmunkálási feladatnak megfelelően alakítsák ki.

Ezek a végpontok a szerszámgép tengelyei, annak ellenére, hogy egyes tengelyek a szerszám-munkadarab relatív haladó mozgásának megvalósítására szolgálnak. A szerszámgép tengelyei adják a szerszám-munkadarab relatív mozgásnak az adott gépen lehetséges szabadságfokait.

Az általunk tárgyalt legbonyolultabb esetben a szerszám és a munkadarab relatív helyzete öt szabadságfokkal meghatározott. A konkrét szerszámgépek szabadságfokainak számát a tengelyek számán kívül befolyásolja az, hogy esetleg nem minden tengely valósít meg önálló szabadságfokot.

Egy szerszámgépet geometriailag meghatározzák a kinematikai lánc szomszédos elemeinek relatív elmozdulási lehetőségei és az egyes elemek

alakja. Ez a szerszám-gép-modell a gyakorlati alkalmazás számára túlságosan bonyolult lenne. Egyszerűsítését szolgálja a munkatéren alapuló megközelítés. Ennek lényege, hogy a mozgás szabadságfokait önmagukban, a szerszám-gépelemek alakjától, azok térbeli elhelyezésétől elvonatkoztatva írják le. Az ilyen modell használatakor ügyelni kell a modellezéskor tett elhanyagolásokra /különösen ütközési, hozzáférési kérdések megoldásakor/.

A szerszám-gép fenti jellemzőinek ismerete még nem elegendő ahhoz, hogy egy adott művelet elvégzéséhez szerszám-gépet, felfogást és készüléket válasszunk.

Ismerni kell például a kinematikai lánc két végpontjának geometriai kialakítását, az egyes tengelyek menti elmozdulások maximális hosszát.

Az egyes tengelyek fontos jellemzője a mozgás sebesség és gyorsulás tartománya és más tengelyekkel való összehangolhatóságának lehetősége.

2.3 Esztergálás

Esztergálásnál a szerszám a munkadarabhoz képest csavarvonal mentén haladó mozgást végez és e haladó mozgással szinkronban a csavarvonal tengelye körül el is fordul /2.7. ábra/. A csavarvonal tengelye egyenes vonal, a pillanatnyi görbületi sugár változhat.

Az esztergálás mozgásait főmozgás és mellékmozgás komponensekre szokás bontani a 2.8. ábra szerint. Ritkán az esztergálás szerszáma egy további tengely körül is elforgatható.

Esztergagépeken az esztergálás főmozgását a munkadarab forgása valósítja meg, a mellékmozgásokat a szerszámnak a munkadarab tengelyén átmenő síkban történő, két egymásra merőleges tengelyre bontott haladó mozgása adja. Ebben az elrendezésben a harmadik térbeli irány /y/ menti haladó mozgás nem megvalósított, aminek következtében a csavarvonal tengelye a munkadarab koordinátarendszerében rögzítetté válik. Esztergagépeken tehát a magmunkált felület /forgástest/ tengelyét a felfogás határozza meg. A mozgásirányok szabványos jelölését a 2.8. ábra mutatja.

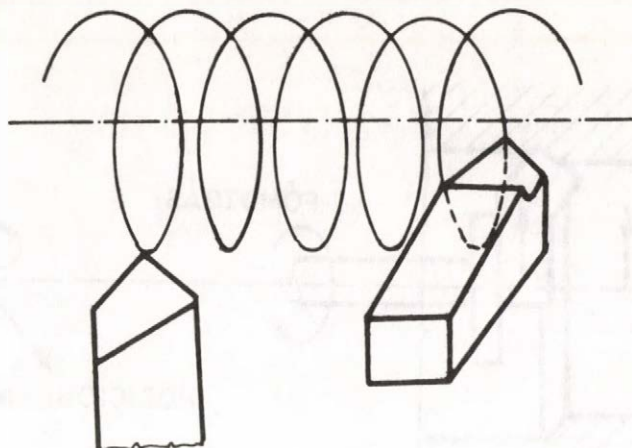
Megmunkáló központokon is gyakran lehetséges esztergálás /2.9. ábra/. Ilyenkor a főmozgást a szerszám forgása adja. A szerszám kinyúlása változtatható /2.10. ábra/, ez a mozgás adja a mellékmozgások egyik ösz-

szetevőjét, a másik összetevő származhat a szerszám, vagy a munkadarab haladó mozgásából. Az esztergálás ilyen kivitele nem teszi lehetővé külső hengerfelületek megmunkálását, az esztergagépekhez képest viszont többletet jelent az, hogy az ábrán x , y és β - val jelölt tengelyek a szerszámgépeken könnyen megvalósíthatók és ezek segítségével a csavarvonal tengelyét a munkadarab koordináta-rendszerében szabadon választ-hatják meg /az xz síkkal párhuzamosan, természetesen/. Ilyen gépeken egyazon felfogásban különböző helyzetű furatok is megtalálhatók.

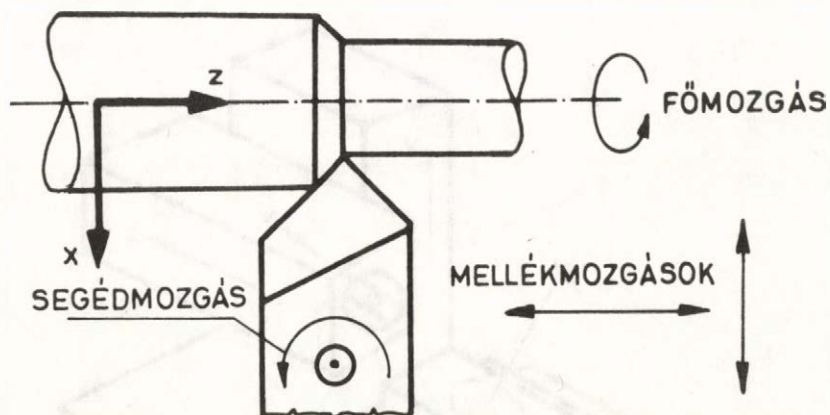
Az esztergálás által kialakított test felületét leírhatjuk úgy, mint a szerszám vágóélei /alkotó/ által az említett csavarvonal menti mozgás során hagyott nyomot. Ezt a felületet olyan hengerszerű felülettel közelítjük, amely egy görbevonalu alkotó rögzített, egyenesvonalu tengely körüli forgatásakor keletkezik /2.11. ábra/. A görbevonalu alkotó két részből tevődik össze: az A2-A3 szakasz a szerszám egymásutáni 360° -os elfordulásaihoz tartozó szerszámélhelyzetek burkológörbéje, az A3-A4 típusu szakaszok pedig a szerszámél közvetlen másolatai. Ezen utóbbi típusu szakaszoknak elsősorban alakos kések alkalmazásakor van jelentősége /2.12. ábra/.

Az A2-A3 szakaszt közelítő burkológörbét úgy határozhatjuk meg, hogy eltekintünk az esztergálás főmozgásától és csak a szerszám xz síkbeli előtoló mozgása során kialakuló görbét tekintjük. Ha a szerszám a csucsnál köralakban lekerekített, akkor az A2-A3 típusu burkológörbeszakasz a lekerekítési középponttal ekvidisztáns görbe lesz.

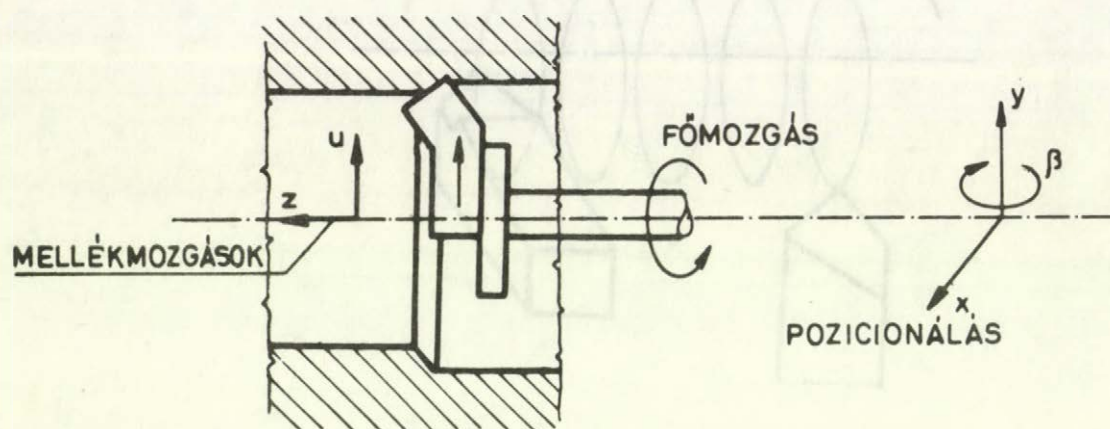
A megmunkált felület alakjának hengerszerű felülettel való közelítésének elvi hibája a szerszám egy fordulatra eső xz síkbeli elmozdulásának mértékétől és a szerszám működő szögeitől / χ elhelyezési szög, ε csucsszög, $\tau = 180^\circ - \varepsilon - \chi$ mellékél elhelyezési szög/ függ. A közelítés jóságát a csavarvonalak h menetemelkedésének szokásosan kis értékei /néhány tized mm/ garantálják /menetesztergálásnál h értéke természetesen nem elvi hiba forrása, hanem a megvalósítandó cél része/.



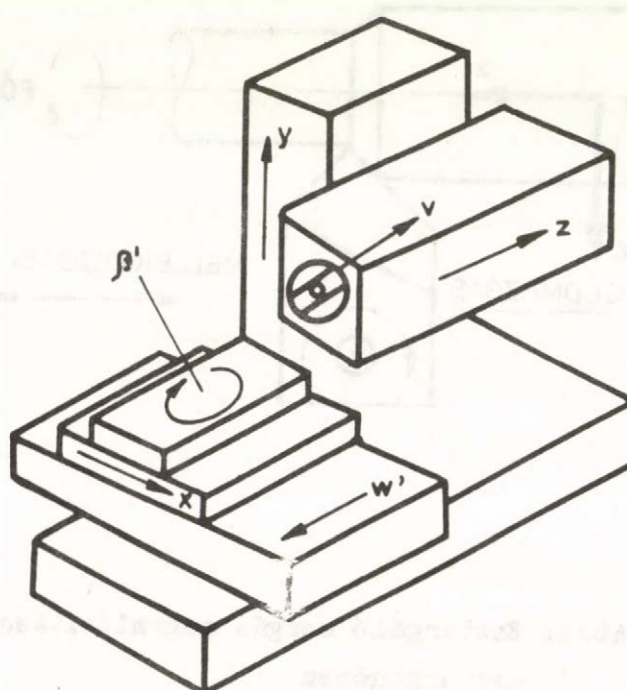
2.7. ábra: Esztergáló mozgás



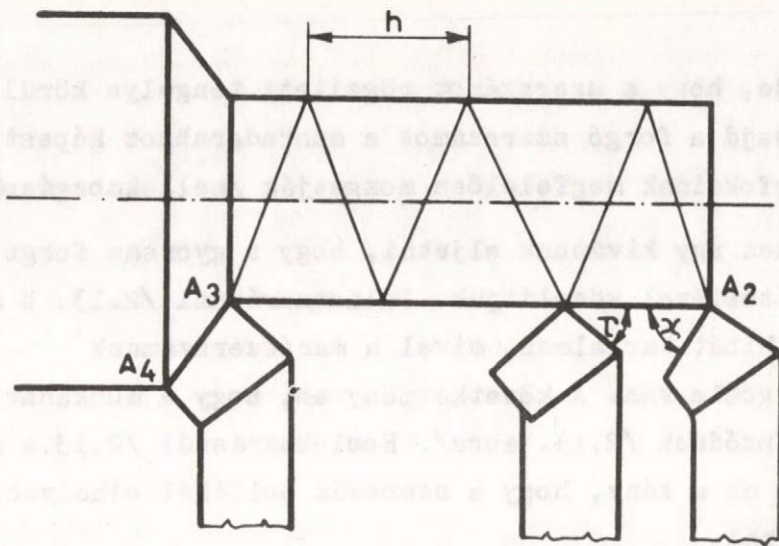
2.8. ábra: Esztergáló mozgás megvalósítása
esztergagépen



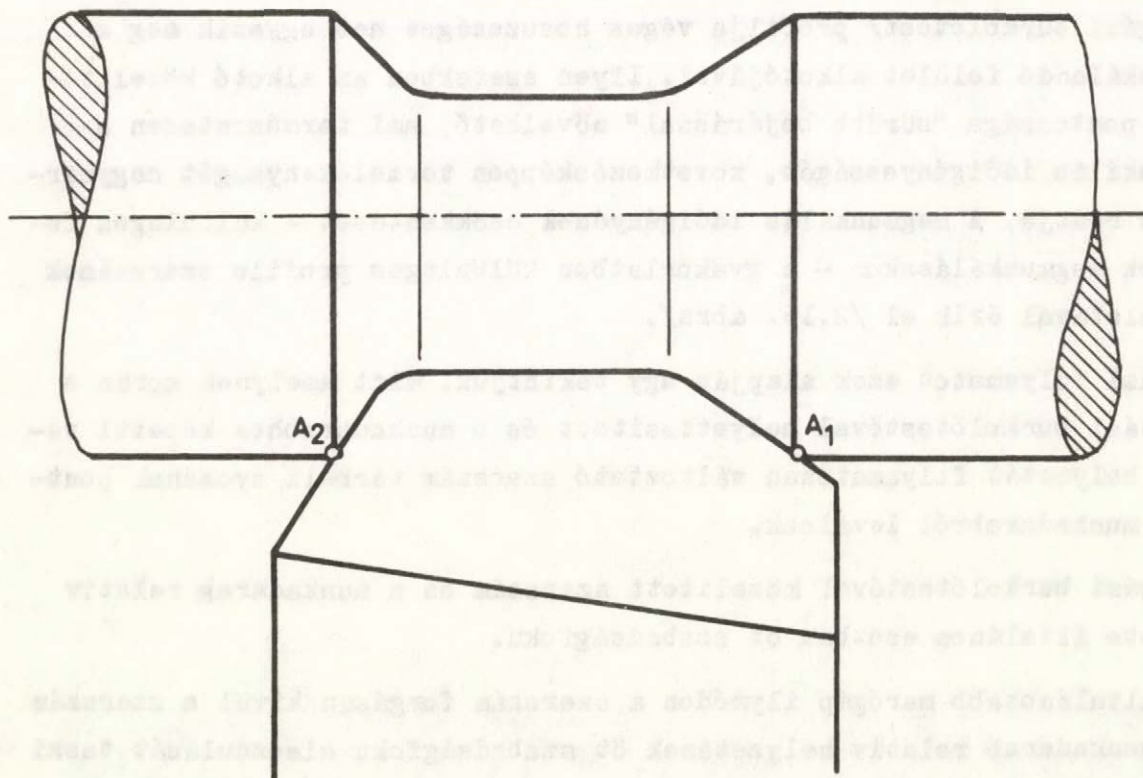
2.9. ábra: Esztergálás marógépen, megmunkálóközponon



2.10. ábra: Megmunkáló központ, kiesztergáló fejjel



2.11. ábra: Esztergáláskor keletkező felület



2.12. ábra: Megmunkálás alakos esztergakéssel

2.4 Marás

A marás jellemzője, hogy a szerszámot rögzített tengelye körül forgatják /főmozgás/, majd a forgó szerszámot a munkadarabhoz képest a szerszámgép szabadságfokainak megfelelően mozgatják /mellékmozgások/.

A marás modelljéhez úgy kívánunk eljutni, hogy a gyorsan forgó szerszámot saját burkolótestével közelítjük. Palástmarásnál /2.13. b ábra/ ez a közelítés elvi hibát tartalmaz, mivel a marószerszámnak csak véges sok vágóéle van. A következmény az, hogy a munkadarab felületén barázdák képződnek /2.14. ábra/. Homlokmarásnál /2.13.a ábra/ az elvi hiba forrása az a tény, hogy a szerszám mellékél elhelyezkedési szöge kisebb 90° -nál.

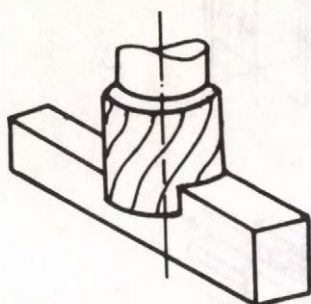
A hibák értéke a szokásos megmunkálásoknál század milliméter nagyságrendű, ezért elhanyagolása megengedhető.

A megmunkálási folyamat elvileg nem lehet pontos akkor, ha a szerszám /-forgási burkolótest/ profilja véges hosszúságon nem egyezik meg a megmunkálandó felület alkotójával. Ilyen esetekben az alkotó közelítésének pontossága "sűrűbb bejárással" növelhető, ami természetesen a megmunkálás időigényességét, következésképpen termelékenységét nagymértékben rontja. A megmunkálás időigényének csökkentését - különleges felületek megmunkálásakor - a gyakorlatban különleges profilu szerszámok használatával érik el /2.15. ábra/.

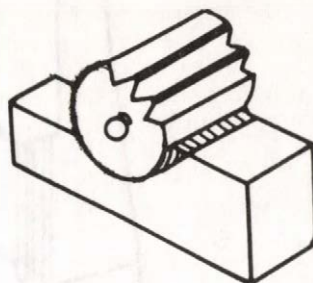
A marási folyamatot ezek alapján úgy tekintjük, mint amelynek során a forgatási burkolótestével helyettesített és a munkadarabhoz képesti relatív helyzetét folyamatosan változtató szerszám térbeli nyomának pontjai a munkadarabról leválnak.

A forgási burkolótestével közelített szerszám és a munkadarab relatív helyzete általános esetben öt szabadságfoku.

A legáltalánosabb marógép ilymódon a szerszám forgásán kívül a szerszám és a munkadarab relatív helyzetének öt szabadságfoku elmozdulását teszi lehetővé /2.16. ábra/.

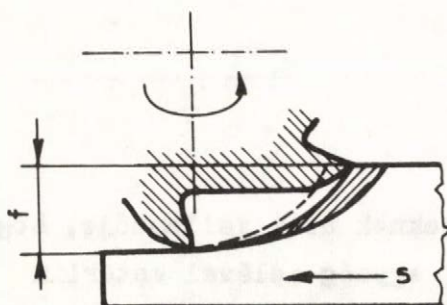


a/ Homlokmarás

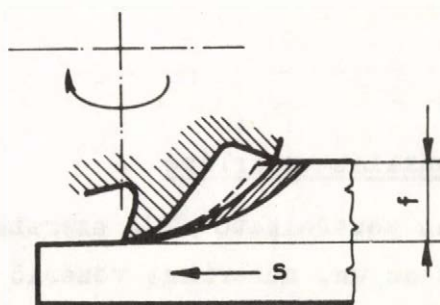


b/ Palástmarás

2.13. ábra: Marás tipikus szerszámai

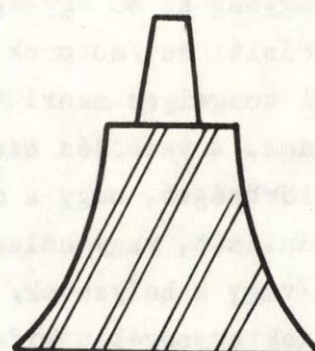
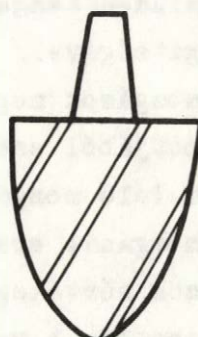


a/ Ellenirányú marás

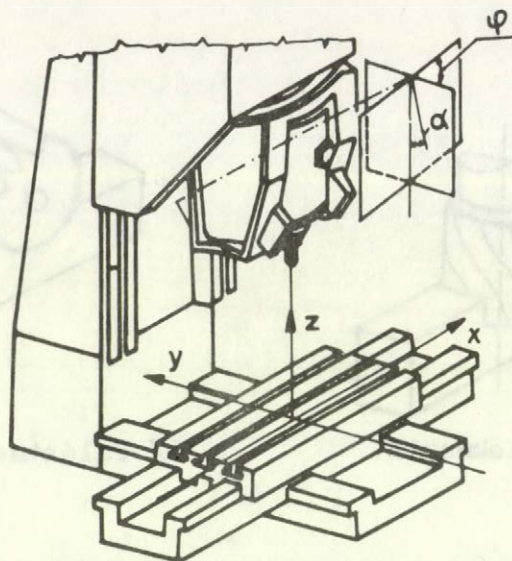


b/ Egyenirányú marás

2.14. ábra: Fogankénti előtolás palástmarásnál



2.15. ábra: Alakos marószerszámok



2.16. ábra: Ötkoordinátás /öttengelyes/ marógép mozgási szabadságfokai

2.5 A megmunkálás vezérlése

A numerikusan vezérelhető /NC/ szerszámgépeknek az a jellemzője, hogy tengelyeiket az un. numerikus vezérlő /NC/ egység jelével vezérlik /2.17. ábra/.

Az NC egység feladata elsősorban az, hogy a vezérlő programban előírt összetett mozgásokat un. pályaszámító egysége /interpolátora/ segítségével a szerszámgép tengelyeinek megfelelő komponensekre lebontsa és minden egyes tengelyre időben folyamatosan generálja a program szerinti mozgásnak megfelelő elmozdulás, vagy pozíció jelét. A mozgató egység az NC egység /minden tengelyre külön szolgáltatott/ jelét feleltet és /motorok segítségével/ a megfelelő mozgásokká átalakítja. A tengelyek szerinti mozgások megvezetettek, rögzített pályán történnek. A vezérlés szempontjából ezért a tengelyek között nem jelent különbséget, hogy a megfelelő mozgás a szerszám-munkadarab relatív elmozdulását, vagy haladó mozgását eredményezi, ugyanis az elmozdulások /vagy a helyzetek/ közös mértékegységben pl. az NC egység kimenő jelének egységében kifejezhetők. A vezérlés szempontjából ezek a mozgások illymódon egyenértékűek egy pontnak az egydimenziós térben történő mozgásával.

A vezérelt pontot a mozgó elemhez /szánhoz/ rögzítve képzelhetjük el. Az NC egység kimenetén a mozgási információ így a szán referenciapontja tengelyekre lebontott mozgásának megfelelő alakban van jelen.

A mozgás komponensek közül külön ki kell emelni a főmozgást. Az esztergálás és a marás főmozgása ugyanis modellünkben nem vesz közvetlenül részt az alakformálásban, csupán sebessége befolyásolja a modell elvi hibáját /kivételt képez a menetesztergálás, de ettől tekintsünk most el/. A főmozgást kapcsolási jelekkel vezérlik, ezekről e pont végén szólnunk.

A vezérléseket pályagenerálási képességeik szerint szokták osztályozni. Az osztályozás alapja az, hogy a vezérlés tudja-e összetett, időben szabályozott mozgások vezérlő jelét generálni /pályavezérlés/, vagy nem /szakaszvezérlés/, és ha igen, akkor egyszerre hány tengely mozgatása lehetséges, milyen trajektória mentén.

Az egyszerre vezérelhető tengelyek száma szerint megkülönböztetnek 2, 3 tengelyes vezérléseket, a pálya alakja rendszerint egyenes és kör, illetve ilyen pályaelemek kompozíciója, a beépített pályaszámító egyestől /interpolátortól/ függően.

A pályamenti mozgás megadása teljes, ha a pálya mellett a pályamenti sebesség értékét is tartalmazza. A vezérlések túlnyomó többsége a pályaelemeket állandó pályamenti sebességgel tudja végrehajtani.

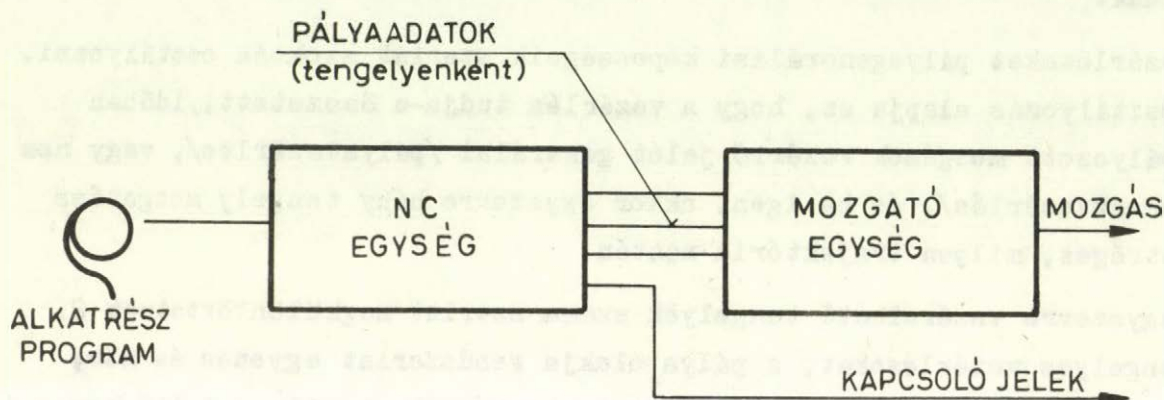
A pályaszámító egység bemenetén tehát a szán referenciapontjának mozgása adandó meg, állandó sebességgel végrehajtandó, adott típusu pályaelemekre lebontott formában. Ezt a formát nevezzük a mozgásmegadás legalsó szintjének.

Térjünk vissza még egy pillanatra a 17. ábrához, az NC egység kimenő jelei között feltüntetett kapcsoló jelekhez. Ezek a jelek különleges mozgásokat vezérelnek /kapcsolnak/, amelyeknek két típusát különböztetjük meg:

- a véges idő alatt lezajló mozgások, amelyek időbeli és térbeli lefutása a folyamat szempontjából közömbös. Ezeknek a mozgásoknak csak a végeredménye fontos, amely többnyire egy ismert, véges állapothalmaz valamely elemének felvétele /ilyen mozgás pl. a szerszámváltás, a körasztal indexelés/.

- olyan mozgások, amelyeket állandósult állapotban nulla vagy egy skálás paraméter határoz meg /ilyen pl. a hűtővíz mozgása, a főorsó forgása/. A mozgások tranziensei a folyamat szempontjából közömbösek.

A kapcsolási információk megadásának módját az NC egységek bemenetén szabványok rögzítik. Ezt a megadási módot tekintjük a kapcsolási információk megadása legalsó szintjének.



2.17. ábra: Numerikus vezérlés

2.6 A megmunkálás pontossága

A tapasztalat azt mutatja, hogy a tényleges megmunkálások során keletkező munkadarab alakja különbözik attól az alaktól, aminek a szerszám-munkadarab relatív mozgás program szerinti végrehajtása esetén keletkeznie kellene. Kihangsúlyozzuk, hogy a referencia alak a programozott mozgásnak megfelelő alak, mivel a vezérléstől ezen alak minél kisebb hibával történő kimunkálását várjuk el. Ezt az alakot a programozás célalakjának, röviden célalaknak fogjuk nevezni.

Az elemzés az alakhibák közvetlenül, vagy számítás, mérés útján kompenzálható, valamint nem kompenzálható komponenseit mutatja ki. Az utóbbi-

ak közé soroljuk az eltérések azon komponenseit is, amelyek a megmunkálás nem mérhető paramétereivel meghatározottak és azokat is, amelyek figyelembe vételét csak a rendelkezésre álló technikai eszközök lehetőségei korlátozzák. Az előbbi alakhibákat szisztematikus hibáknak, az utóbbiakat sztochasztikus hibáknak nevezzük. Szisztematikusnak tekintjük azokat az alakhibákat is, amelyek rendszeres jellegét tapasztalati úton, a készdarabok mérésével derítik fel.

A sztochasztikus hibák csökkentése technológiai, konstrukciós stb. probléma, tárgyunkon ezért kívül esik. A sztochasztikus hibákat a megmunkálás pontosságával jellemzik.

A szisztematikus hibák lehetnek a konkrét munkadarabból függetlenül is kompenzálhatók, vagy olyanok amelyek kompenzálásához a munkadarabot, a technológiát is ismerni kell. Az első hibafajta számunkra érdektelen, mivel a kompenzáció az alkatrészprogramozáson kívül eső feladat. A második hibafajtaival kívánunk részletesebben foglalkozni.

A szisztematikus hibákat is a numerikusan vezérelt forgácsolás kifejtett modelljében tárgyaljuk. Ebben a modellben az alakhibák három jelenséggel magyarázhatók:

- a mozgás pályája eltér a programozott pályától;
- a szerszám alakja a megmunkálás során változik;
- a mozgás sebessége nem azonos a tervezett értékkel.

Az első két jelenség a készdarab alakját közvetlenül módosítja. A sebesség változásának közvetlen hatása az, hogy megváltozik modellünk említett elvi hibája. Ettől a közvetlen hatástól azonban a továbbiakban eltekintünk.

A mozgás tényleges pályája, a szerszám alakváltozása, a mozgás sebessége bonyolult kölcsönhatásban áll egymással és a megmunkálás egyéb paramétereivel /forgácsoló erő, forgács, szerszám stb./.

A tényleges megmunkálási folyamat modelljéhez úgy kívánunk eljutni, hogy ezektől a kölcsönhatásoktól elvonatkoztatva az eredő szisztematikus megmunkálási hibákat két tényezőre vezetjük vissza: a szerszám alakváltozására /kopására/ és a szerszám-munkadarab relatív mozgás pályájának torzulására.

Ez a modell a vezérlésekkel szemben azt a követelményt támasztja, hogy a programozott névleges pályát a szerszám kopása és ismert pályatorzulások ellenében is tartani tudja.

A szerszám kopásából eredő alakváltozást a számításigény ésszerű korlátok közé szorítása érdekében a kinyulás /hossz/ és szerszám sugár fogalmaival közelítjük. Az ilyen adatokkal jellemzett szerszám-kopás korrigálásához elvileg elegendő a kopás mértékének és a névleges pályának az ismerete. A következő pontnak elébe vágva azonban megjegyezzük, hogy számítási teljesítményük korlátozottsága miatt a mai vezérlések előre kiszámított, összetett pályaadatok megadását igénylik.

A mozgáspálya korrekció elvégzéséhez a vezérlésnek ismernie kell a korrigáló relativ pályát. Az általános eset megoldása irracionális követelmény. Ésszerű korlátozást jelent a korrigáló pálya alakjának behatárolása. A vezérlés tehát eleve "ismer" bizonyos korrigáló pálya típusokat /mint ahogy a pályaszámító egység is csak bizonyos típusu pályaelemeket ismer/. A vezérlő program a korrigáló pálya típusát és konkrét paramétereit tartalmazza.

2.7 Technológiai tervezés, alkatrészprogramozás

Végezetül rátérünk az alkatrészprogramozással kapcsolatos fogalmak ismertetésére. Ez az ismertetés részben a már említett tények összefoglalása.

A vezérlő program az NC egység működését írja elő. Összeállításának folyamata az alkatrészprogramozás. Az alkatrészprogram megírásának fő kiindulási adata a nyersdarab alakja és a készdarab konstrukciós terve.

A készdarab konstrukciós terve a készdarab lehetséges alakjait definiálja. Az alkatrész konstrukciós tervet egy névleges konstrukció, tehát az alkatrész egy lehetséges alakja és a névleges konstrukció paramétereire előírt tűrések segítségével lehet megadni.

Ez a felületes megjegyzés a témát távolról sem meríti ki, a további kifejtésre azonban ehelyütt nem vállalkozhatunk.

A konstrukciós terv által előírt alakhalmaz és a nyersdarab alakja meghatározza az anyageltávolítási feladatot, a tulajdonképpeni forgácsolá-

si feladatot. A forgácsolási feladat tehát egy alakhalmaz, amely a nyersdarab halmaz és a készdarab halmaz minden egyes eleme közötti regularizált halmazelméleti különbség eredménye.

Az alkatrészprogramozás első lépése a feladat bonyolultságától függően a részfeladatokra bontás.

A megmunkálási feladat azon részfeladata, amelyet a munkadarab felfogásának változtatása nélkül lehet elvégezni, a technológiai művelet. A műveletek műveletelemekre bomlanak.

A műveletelem olyan forgácsolási feladat, amelyet egy szerszám folyamatos munkával végez el. A műveletelem általában összefüggő anyag-rész eltávolítását is jelenti, bár ezalól a szabály alól kivételek is vannak. A műveletelemekre bontás során először általában egy vagy több, nagyobb variációju /kisebb pontosságu/ feladat /nagyoló műveletelem/ határozható meg és csak az utolsó feladat variációja /pontossága/ írja elő a konstrukcióban megadott értéket.

Az alkatrészprogramozás eddig említett lépései a műveleti sorrendtervezés és művelettervezés. A sorrendtervezés és művelettervezés eredménye a megmunkálás műveletelemeinek leírása, ide értve az egyes műveletelemek sorrendiségére tett megkötéseket és a műveletek, műveletelemek végrehajtására alkalmas technológiai elrendezés /felfogás, szerszámgép, szerszám/ megválasztását.

Megjegyezzük, hogy a műveleti sorrendtervezés és művelettervezés az alkatrészprogramozásnak talán a legnehezebben formalizálható, és ezért legnehezebben automatizálható fázisa. A fejezet előző pontjaiban már megemlítettünk olyan problémákat, amelyek megoldása a sorrendtervezés formalizálásához nélkülözhetetlen.

A következő lépés a műveletelemek részletes technológiai tervezése. Ennek első lépésében minden egyes műveletelemre a megmunkálás sztochasztikus hibáinak figyelembevételével meghatároznak egy céltestet. Ez a céltest lesz a pályaszintézis vagyis a műveletelem-tervezés fő kiindulási adata.

A műveletelem-tervezés számára a forgácsolási feladat tehát ráhagyási alakzat formájában adott. A műveletelem-tervezés feladata olyan szerszám-munkadarab relatív mozgás megtervezése, amely egyrészt adott szerszámgépen realizálható, másrészt amelynek során a forgácseltávolítási

feladat megoldást nyer. Ezen mozgás programja a numerikus szerszámgép-vezérlés vezérlő programja.

A műveletelem tervezési feladat ebben a formában határozatlanságot tartalmaz, amelyet a műveletelem-tervezés inputján megadott további adatok: a szerszámgép, szerszám befogás, megmunkálási mód, valamint egyéb technológiai /fogásmélység/ és gazdasági /üresjáratok/ kötöttségek oldanak fel.

A mai vezérlő rendszerek vezérlő programjai közvetlenül a forgácsoló mozgásokat specifikálják. A mozgások mélyebb értelme azonban mindig az a ráhagyási alakzat, amelynek eltávolítására az adott mozgást tervezték.

A következő fejezetnek elébe vágva megjegyezzük, hogy a vezérlő nyelvekben a mozgást közvetlenül leíró, alacsony szintű nyelvi primitivек helyét fokozatosan egyre magasabb szintű primitivек foglalják el. Az utóbbi időben ez a tendencia gyorsuló ütemben érvényesül és már kirajzolódik annak távlati lehetősége, hogy a vezérlési nyelven közvetlenül a ráhagyási alakzatot programozzák. Ez a folyamat indokolja, hogy a forgácseltávolítást éppen a fenti módon modellezzük.

Végezetül megemlítjük a kapcsolási jellegű információ feldolgozásának kérdését. A műveletelem tervezés inputján illetve a tervezés folyamatában a kapcsolási információ szerszámgéptől független, funkcióra és eredményre orientált formában keletkezik. A tervezési-vezérlési folyamat outputja az a szerszámgép- illetve vezérlésfüggő forma, amelyet a kapcsolási információ megadása legalacsonyabb szintjének nevezünk.

3. A numerikus szerszámgépvezérlő rendszerek és az alkatrészprogramozási módszerek fejlődése

A felmérés elsődleges célja a numerikus szerszámgépvezérlő rendszerekben alkalmazott alkatrészprogramozási módszerek áttekintése és kritikai értékelése.

A felmérés fókuszában az NC berendezések vezérlési nyelvei állanak. Ezeket a nyelveket azonban nem vizsgálhatjuk önmagukban. A fejlődés az NC berendezések általános rendszertechnikai kialakításának, szolgáltatásrendszerének fejlődésével összhangban ment végbe. A fejezet első része ezért ezt a környezetet vizsgálja.

A vezérlési nyelvekre a gyártó rendszerek megjelenése, az integrálási törekvések is hatással vannak. Szükséges ezért a gyártórendszerek létrehozása terén elért eredményeket is tárgyalni.

A második rész közvetlenül a vezérlési nyelveket kíséri meg összehasonlítani. Célunk a jövő trendjének megjósolása a jelen helyzet alapján, ezért a múlt már túlhaladott eredményeit csak futólag, a teljesség igénye nélkül említjük. Ha az NC berendezésekben már megvalósított legbonyolultabb funkciókat keressük, akkor elegendő a számítógépes szerszámgépvezérléseket figyelembe venni. A fejezetben elsősorban ilyen rendszerekre koncentrálunk.

Az NC berendezésnek a technológiai tervezéssel való kapcsolata a vezérlési nyelv szempontjából különösen lényeges tényező, mivel az alkatrészprogram szinte teljes egészében a technológiai tervezés outputja.

A hagyományos NC berendezések vezérlési nyelveit a megvalósítható funkciók mennyisége korlátozza.

A számítógépes szerszámgépvezérlések egyre több olyan funkciót ellátnak, amelyek azelőtt a technológiai tervezés alsó szintjeihez voltak rendelve [12, 143]. E fejezet különböző részeiben ezért a teljesség igénye nélkül utalunk arra, hogy a vezérlési nyelvek processzorai hogyan egészítik ki a vezérlések képességeit. Ezen processzorok teljes felmérése igen messzire vezetne, ezért tulajdonságaikat csak az említett szempontból tudjuk értékelni.

3.1 Az NC rendszerek rendszertechikája, szolgáltatásrendszere

3.1.1 A fejlődés főbb állomásai

A gépgyártás automatizálásának nagy jelentőségű fejleménye volt a számjegyes szerszámgépvezérlések /NC berendezések/ megjelenése és ipari alkalmazása. Az első NC berendezésekről az 1950-es évek elején tudósítottak. Az azóta eltelt időben a századunkra oly jellemző módon ez a technika is gyökeres változásokon és óriási fejlődésen ment keresztül. Ezzel párhuzamosan gyors ütemben nőtt az iparban alkalmazott NC berendezések abszolút száma és részaránya az összes szerszámgépeken belül [1, 2, 3, 4, 5]. A telítettségről szóló feltételezések [5] ellen szól az árak csökkenése miatt várható alkalmazási terület-bővülés [4, 6, 7]. Ebben a fejezetben röviden leírjuk az NC technika fejlődésének fő állomásait és áttekintjük a jelenlegi rendszerek jellemzőit.

Az első idők NC egységei huzalozott logikájú vezérlések voltak, ezeket hagyományos NC egységeknek fogjuk nevezni. A hagyományos NC egységek idejében alkalmazható eszközök behatárolták az NC egységbe építhető funkciók mennyiségét és megkötötték a rendszerek alkatrészprogramozási nyelvét kidolgozó tervezők kezét.

Döntő fordulatot hozott a számítógép betörése az NC technika területére. A lehetőséget erre a fordulatra a számítástechnikai berendezések építőelemeinek ugrásszerű, mondhatni forradalmi fejlődése teremtette meg. A számítógépeket tartalmazó új vezérlési rendszereket új elnevezéssel illették: CNC, DNC rendszereket fejlesztettek ki. Ezek az elnevezések a vezérlés belső architektúrájára utalnak:

- a hagyományos NC funkciókat, illetve azok továbbfejlesztését megvalósító, számítógépet tartalmazó rendszereket CNC[≡] rendszereknek,
- a szerszámgépcsoportot vezérlő, a csoport elemeinek munkáját összefogó, összehangoló, felügyelő, kiszolgáló rendszereket DNC[≡] rendszereknek nevezik.

≡ CNC: Computerized Numerical Control
Számítógépes számjegyes vezérlés

≡ DNC: Direct Numerical Control
Közvetlen számjegyes vezérlés

Az NC technika széleskörű elterjedésével egyidőben nyilvánvalóvá vált, hogy az ilyen berendezések sikeres alkalmazásához, lehetőségeik teljes kihasználásához szükség van a gyártás kapcsolódó folyamatainak számítógéppel való segítésére, automatizálására.

Az olyan rendszereket, amelyek az NC, vagy DNC rendszereknél lényegesen több feladatot látnak el, CAM /Computer Aided Manufacture - számítógéppel segített gyártó/ rendszereknek nevezik. A CAM rendszereket a DNC rendszerektől elválasztó határvonalat természetesen nehéz meghuzni. A CAM rendszerek a közvetlen megmunkálási folyamat vezérlésén kívül a gyártástervezés, - irányítás, szállítás, mérés stb. egyes részfolyamatait segítik vagy automatizálják [14, 15] .

A CAD/CAM rendszerek a konstrukciós tervezést is egyesítik egy nagy rendszeren belül /CAD: számítógéppel segített tervezés - Computer Aided Design/.

Az alábbiakban, a történeti sorrendet követve elemezzük a legtipikusabb megoldásokat.

3.1.2 A csoportos vezérlések

A gépipar, mint a számítógéptechnika új alkalmazási területének meghódításakor szinte azonnal kiformálódott a távlati cél: a gépgyártás minél több folyamatának automatizálása egy rendszeren belül. Az irodalom kezdettől fogva integrált gyártórendszerekről, integrált adat- és anyagfeldolgozó rendszerekről, számítógépes tervező- és gyártórendszerekről /CAD/CAM/ tudósított [6, 16, 17, 18, 19, 20, 21, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139] . Ezen rendszerek egy alrendszere a számítógéppel csoportosan vezérelt szerszámgépcsoport, amelyben a számítógép szabad kapacitását a közvetlen megmunkálás vezérlésén kívül más folyamatok automatizálására, illetve az azokat automatizáló más alrendszerekkel való együttműködésre lehet felhasználni.

Éppen ez a többlet az, ami egy vezérlési rendszert perspektivikussá, a csoportos vezérlést, vagy az egyedi vezérlések csoportos működtetését pedig távlatilag szükségsszerűvé teszi. A csoportos vezérlések tárgyalásában ezért ezekre a közvetlen vezérlési funkciókon túlmenő szolgáltatásokra koncentrálunk.

A 60-as évek végén a számítógéptechnika fejlettsége a csoportos vezérlések /DNC rendszerek/ kialakulásának kedvezett [22] /az 1970-es Chicagói Szerszámgépkiallításán például több ilyen rendszert mutattak be/.

Ezen időszak rendszerei közül meg lehet említeni a General Electric Comman DIR rendszerét, [22], amely egy tipikus BTR /behind the reader - a lyukszalagolvasó megkerülésével/ rendszer. Az egyes szerszámgépek mellett megtaláljuk a számítógéphez csatlakoztatott hagyományos vezérlést, és mindegyik gép mellett van egy kezelői állomás a szükséges csatoló logikával. A kezelői állomás teszi lehetővé a kezelő és a műhely vezérlését segítő számítógép közötti on-line információ cserét. Ez a lehetőség hagyományos NC berendezések esetében csak közvetve, az automatizált rendszer megkerülésével létezett.

Ugyancsak BTR rendszerű a Fujitsu cég Fanuc System-K elnevezésű rendszere, amelynek párja, a Fanuc System-T az ellenkező véglettel: teljes software interpolációval készült [23, 24, 25]. A System-T elrendezésben az egyes szerszámgépek mellett minimális elektronikát tartalmazó MTC /Machine Tool Controller/ egységek találhatók. Egy számítógéphez egyidejűleg T és K rendszerben vezérelt szerszámgépek is csatlakoztathatók. A rendszer igen lényeges eleme a nagy kapacitású mágnesdobos háttértároló, amelynek segítségével nagyon jól kezelhető software rendszert építettek ki. A műhely munkáját a számítógép feladatlista alapján irányítja, az alkatrész programokat a dobon tárolt programkönyvtárból hívja le. A munka menetéről helyzetjelentést lehet kérni. /Érdekességgként meg kell jegyezni, hogy a diszpécser szolgáltatások csak a kibővített software rendszerben léteznek, új feladatlista bevitel és a pillanatnyi helyzet lekérdezése a kibővített rendszerben is csak a szerszámgépek állásidejében /off-line üzemmódban/ lehetséges. /A feladatlista módosítása on-line, bármikor kérhető/.

A software rendszer lehetővé teszi a tárolt programok tetszőleges módosítását, alapkiépítésben off-line, kibővített rendszer esetén on-line üzemből.

Nagyon fejlett a System-T szubrutin programozási rendszere. A szubrutinokból technológiai könyvtárat lehet kiépíteni.

A System-T és System-K rendszerszoftware-en kívül a vezérlő gépre egyéb programok is készültek. Ilyenek például a párbeszédés FAPT alkatrészprogramozási nyelv, az alkatrészprogram ellenőrző és rajzoló program, és az adatformátum konvertáló program. Az utóbbi a különböző kódrendszerben írt programok konverzióját végzi a System-T adatformátumára.

A System-T és System-K rendszerek szolgáltatásai igen széles körre terjednek ki, amelyek feltehetőleg még tovább is bővíthetők. Ezzel alapozhatták meg a rendszer nagy sikerét: 1967-1972 között mintegy 40 rendszert helyeztek üzembe.

Erre az időszakra jellemző az Actron Ind. TRIDEA rendszere is, amelyet ettől lényegesen eltérő koncepció alapján fejlesztettek ki [26, 27].

A TRIDEA vezérléseket új egységek beállításához, régiak korszerűbb egységekkel való lecseréléséhez és régi egységek módosításához tervezték.

A teljes TRIDEA DNC rendszer három szintű összekapcsolt számítógéprendszert tartalmaz. A legfelső szinten levő nagy gép az alkatrészprogramok generálásával, tárolásával és módosításával kapcsolatos feladatokat végzi. A középső szint gépe 1-10 szerszámgép számára végzi az interpolációt és ellátja a csoport felügyeletét. A szerszámgépek mellett álló egységek minigépet tartalmaznak és a közvetlen vezérlésen kívül az adatgyűjtéssel kapcsolatos funkciókat is ellátják. A rendszerhez a nagygéppel összeköttetésben álló, hordozható katódsugárcsöves adatátlomlás is tartozik.

A DNC rendszer helyi vezérlőegységéhez hasonló kivitelben önálló CNC egység is készült. Ennek előnye, hogy alkalmazása esetén a vezérlés később könnyen integrálható a DNC rendszerbe.

A TRIDEA rendszer sokoldalúsága egyrészt mutatja a számítógépes szerszámgépvezérlés egységességét, másrészt jól jellemzi az akkori hardware lehetőségeket.

3.1.3 Az egyedi vezérlések

Ha a további fejlődés tényeit vizsgáljuk, akkor meg kell állapítanunk, hogy a számítógépes szerszámgépvezérlések területén a csoportos vezér-

lések mellett nagy számban jelentek meg egyedi számítógépes szerszám-gépvezérlések /CNC rendszerek/ [10, 11, 28, 29, 30, 31, 32, 33, 34, 35].

Tekintve, hogy a CNC nem egyéb, mint a technika mai színvonalán tervezett és gyártott egyedi számjegyes vezérlés, ez a kép - noha új, nem váratlan.

Mik voltak ezen fejlemény okai ?

A csoportos vezérléssel szemben alapvető elvárás, hogy a hagyományos vezérlési funkciók ellátásán túl a rendszer más folyamatok vezérlését, automatizálását is ellássa. Ezen folyamatok automatizálása azonban ma még részben kezdeti stádiumban van /pl. mérés, szállítás/ [127] vagy ezeket minden egyes installációban egymástól gyökeresen különböző algoritmus szerint kell megoldani [4, 36]. /Pl. termelésirányítás, adatgyűjtés./ Ilyen körülmények között általános rendszert létrehozni nehéz feladat.

A csoportos vezérlések software szolgáltatásai rendszerről rendszerre különbözőek. Ezek a szolgáltatások érintik leginkább az üzem meglévő szervezetét, ami a feladat nehéz megfogalmazhatóságával és többletberuházásokkal jár [37, 38, 140, 141, 142].

Gazdasági ok, hogy csoportos vezérlés létesítése nagyszámú beruházást igényel, amelynek megtérülését érzékenyen befolyásolja a rendszer kihasználtsága, rendelkezésre állása. A kihasználtságot döntően befolyásolja a termelésirányítási rendszer eredményessége, amelyet nagyon nehézkes mindjárt az első perctől kezdve maximális szintre hozni.

A CNC rendszerek elterjedésének másik oka a minigépek rohamos fejlődése és a legutóbbi időben a mikroszámítógépek megjelenése [39, 40]. Az árak és a méretek csökkenésével [42, 43] az egyedi vezérlések egyre több új, bonyolultabb feladatot tudtak ellátni, olyanokat, amelyek, azelőtt csak a DNC rendszerek sajátosságai voltak [8, 10, 28, 30].

Mindez oda vezetett, hogy jelentős kereslet keletkezett olyan vezérlések iránt, amelyek egyedi gépvezérlési funkciókat magas színvonalon képesek ellátni.

3.1.4 CNC berendezések jellegzetességei

Az alábbiakban a viszonylag újabb keletű berendezések szolgáltatásait kíséreljük meg áttekinteni [10, 11, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56]. Vizsgálatunk az alábbi területekre terjed ki:

- vezérlési funkciók
- kezelői interface
- szerszámgép interface
- karbantartás, hibaelhárítás.

Témánk szempontjából leglényegesebbek a vezérlési funkciók. Tekintettel arra, hogy ezek közvetlenül tükröződnek a vezérlő nyelvben, ismertetésükre külön pontot szánunk /3.2 pont/.

Nem tartozik közvetlenül a vezérlési funkciók közé, de azokat nagymértékben meghatározza a vezérlő rendszerek programtárolási képessége. A félvezető technika fejlődése következtében ugyanis ma már egyedi vezérlésekben is gazdaságosan megoldható a vezérlő programok operatív tárban való tárolása. Ez a megoldás növeli a megbízhatóságot /elmarad az alkatrészprogram beolvasásának szükségessége minden egyes munkadarab legyártásánál/, és lehetővé teszi a programok lyukasztás nélküli javítását, módosítását.

Említést érdemel /bár nem CNC !/ a Sperry-Vickers cég UMAC vezérlése [57]. Ehhez opcióként többlet tárolókapacitást lehet választani /ferrit-tár/, vagy nagyobb igények esetén mágneslemezes tároló /!/ beépítése is lehetséges /kb. 120 Kbyte tárolására/.

A programszerkesztéssel kapcsolatban megjegyezzük, hogy teljesen általános megoldás realizálása ma még meglehetősen drága, ezért nem vesztették el jelentőségüket a speciális programszerkesztő-ellenőrző berendezések sem [65, 66, 67, 68].

Egyes alkalmazásokban a program gyakori módosításának különleges jelentősége van [52, 64].

A vezérlő nyelvben közvetlenül nem tükröződik, de figyelemre méltó egyes vezérlések azon képessége, hogy a vezérlő számítógépek szabad kapacitását adatgyűjtésre, a munkamenet értékelésére használják /pl. CIT ALCATEL CN741, ALLEN-BRADLEY 7300/. Szerszámélettartamot, sorozatnagyságot figyel a SIEB-MEYER CNC 25-05 vezérlőegysége.

A számítógép megjelenésével nagymértékben egyszerűsödött a vezérlések kezelői interface-je. Eltűntek a korrekciók beadására szolgáló drága és viszonylag kevésbé üzembiztos peremkerekek, helyüket számbeviteli tasztatura váltotta fel. A kijelzéshez többhelyütt katódsugárcsőves alfanumerikus display-t használnak, bár ezen gyakran igen zsufoltan jelenít meg a többlet-információt. Megjelentek a félvezetős, viszonylag kis kapacitású alfanumerikus display-ek, és ehelyütt meg kell említeni a magyar fejlesztés uttörő vonását.

Ritkábban, a jobb hozzáférhetőség érdekében, a tasztaturát és displayt a szerszámgépre kihelyezett egységbe telepítik [72].

A számítógépes vezérlésekben ujabban gyakran elmaradnak a szerszámgépek mellett eddig alkalmazott szekvenciális relés logikák [69, 70, 71].

A relés logika funkcióit egyes esetekben software uton /CIT ALCATEL CN741, Allen Bradley 7300, Bendix Dynapath System 5, stb. [73]/, vagy ugynevezett programozható vezérlőberendezések /PC-Programmable Controllers [74]/ beépítésével valósítják meg /Mark Century 1050 és még sokan mások/. Utóbbiak olyan egységek, amelyeknek logikai és relés bemenetei, valamint logikai és relés kimenetei vannak, és a működés algoritmusát valamilyen félvezetős tároló tartalmazza. A tároló cseréjével vagy újraprogramozásával a működés algoritmusát módosítható.

A számítógépek elterjedésével nagymértékben javult a szerszámgépvezérlések karbantarthatósága [75].

A CNC rendszerekben általában már az alap vezérlő software is tartalmaz ellenőrzéseket, amely az észlelt hibák kódját üzemközben rögtön meg is jeleníti. A megjelenítés gyakran szöveges üzenet formájában történik /a CIT ALCATEL 741-ben külön opció az üzenetek választéka/, másutt csak a hagyományos lámpák kigyújtásával történik.

A hibák elhárításához rendszerint az alap vezérlőprogramot leváltó speciális ellenőrző programokat adnak, amelyeket a vezérlésbe beolvasva a hibakeresés esetleg elemszintre lebontva is elvégezhető.

A drága szerszámgepeket vezérlő bonyolult vezérlések számítógépes karbantartásának jövőbe mutató példája a Kearney and Trecker hibafelderítő kommunikációs rendszere. Ebben a rendszerben meghibásodás esetén a gyártó speciális szerszámgépe közönséges telefonvonalon on-line a kérdéses rendszerhez kapcsolható.

Megállapítható, hogy a számítógépes hibafelderítésnek döntő hatása lesz a rendszerek üzemben tarthatóságára.

A fenti vázlatos felsorolás bebizonyíthatta, hogy a számítástechnika fejlődésével az egyedi vezérléseken belül több bonyolult vezérlési funkciót tudtak gazdaságosan ellátni. Egyszerűsödött a kezelés, javult a karbantarthatóság.

A számítógépes egyedi vezérlések terjedése a hagyományos NC rovasára várhatóan gyorsulni fog és így a csoportos vezérlésekkel szembeni igények kiformálódásához is hozzájárul.

3.1.5 A csoportos vezérlések fejlődése

A csoportos vezérlések vonzó lehetőségei miatt a kutatások világszerte tovább folytatódnak, és újabb bonyolult rendszereket helyeznek üzembe [4, 38, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86].

Az alábbiakban példaképpen felhozott rendszerek a folyamatok integrálásának egy-egy megközelítését illusztrálják. A példák talán jól megvilágítják az integráció folyamatának nehézségeit, ugyanakkor éreztetik azokat az előnyöket, amelyekhez az ilyen rendszerek alkalmazói a termelékenység növekedésével juthatnak.

Az integrált rendszerek felé vezető egyik utat a Kearney and Trecker cég flexibilis megmunkálási rendszerével illusztrálhatjuk [87, 88, 89]. Ez a rendszer NC gépeket, automatikus anyagmozgató berendezéseket és számítógépeket tartalmaz.

A fő cél az alkatrészválaszték egyes darabjainak random sorrendben történő flexibilis megmunkálása, az átállási idők csökkentése. A rendszerben bonyolultabb megmunkálóközpontok is alkalmazhatók, kihasználásukat a rugalmas vezérlés biztosítja. A konkrét megmunkálógépeken kívül a rendszer tartalmazza a nyersdarabok megmunkálás előtti osztályozásához

szükséges eszközöket, továbbá tisztító és mérőgépeket. A szállítóberendezés az alkatrészválaszték minden darabját közvetlenül mozgatni tudja.

A megmunkálási rendszer vezérlését /az egyes egyedi NC vezérléseken kívül/ három számítógép látja el. A DNC számítógép a vezérlési programokat kezeli és jelentéseket készít, a másik két gép vezérli a szállítóberendezést, kezeli a szerszám paletta és munkadarab nyilvántartását /állapot és helyzet/, jegyzi a szerszámok élettartamát és elvégzi a termelésirányítási feladatot.

Egy másik szerszámgépgyár hasonló integrálási törekvéséről számol be [90]. A bonyolult, nagy pontosságú megmunkálóközpontokkal /Olivetti Auctor és Horizon család/ a vezérlést már korábban is együtt szállítottak /CN5D/CNC/, az igény azonban az, hogy az ilyen rendszerekhez alkatrészprogramozási rendszert és méretellenőrzési rendszert is biztosítani tudjanak. A kidolgozott programnyelv /GTL/ szimbólikus geometriai leírást, és típusalakzatok /furatok, zsebek/ lehívását teszi lehetővé. A rendszer tervezésénél lényeges szempont volt a gyártáselőkészítés megkönnyítése: a processzor szerszámigény listát, szerszámelrendezési tervet generál.

A műhely alkatrészprogram ellátására kidolgozott eszközöket integrálja egy rendszerbe az IBM 5275 DNC rendszere [91]. Az IBM DNC rendszer fő célkitűzése, hogy megkönnyítsék és meggyorsítsák az alkatrészprogramok írását, tárolását, javítását és a szerszámgépekhez való eljuttatását. A hardware rendszer új eleme egy kerekeken guruló, katódsugárcsöves kijelzővel, tasztatúrával ellátott berendezés, amelyik egyik oldalon az IBM nagyszámítógéphez, másik oldalán pedig max. 2 NC szerszámgép vezérléshez csatlakoztatható. A berendezés ezenkívül off-line módban önállóan működtetve is használható.

A rendszert nagyszámítógépes üzemmódba kapcsolva forrásnyelvű alkatrészprogramokat lehet a nagyszámítógépbe küldeni, fordítást kezdeményezni /processzálast-posztprocesszálast/, lefordított alkatrészprogramokat /vezérlőprogramokat/ visszakérni. Off-line üzemmódban, nagyszámítógép nélkül a szerszámgépekkel összekövet vagy anélkül alkatrészprogramok javítása, módosítása végezhető. A rendszer kezelése a funkcióknak és nem a számítógépek sajátosságainak megfelelő nyelvezettel történik. A mozgatható szekrény lehetővé teszi, hogy az egész műhely alkatrészprogram ellátását néhány ilyen terminállal ugrásszerűen megjavítsák.

A rendszerhez tehát teljes software készült, beleértve a nagygéppel való kapcsolathoz szükséges programokat is. Fő előnye megint az, hogy saját területén teljes, kerek szolgáltatásrendszere van.

Számítógépes gyártórendszerek létrehozására a szocialista országokban is széleskörű kutató-fejlesztő munka folyik, amelynek eredményeképpen már több, különböző rendeltetésű rendszert helyeztek üzembe [36, 82, 92, 93, 94, 95, 96, 97, 99]. A kifejlesztett rendszerek többsége BTR rendszer [106, 107, 138, 139], eltérő felépítésű a következő pontban ismertetett magyar rendszer. A gyártórendszerek központi számítógépe alkatrészprogram könyvtárat kezel, a termelésirányítást informálja és segíti, szállítást, raktározást vezérli. Az alkatrészprogramok szerkesztése, javítása, módosítása is a központi gépek kapcsolódó perifériáiról lehetséges [98].

A csoportos vezérlések legújabb fejleményei között feltétlenül külön meg kell említeni a japán eredményeket [4, 21, 25, 44, 78, 79, 80, 83 107].

Pillanatnyilag ugyanis ezek a rendszerek terjedtek el leginkább. A japán rendszerek sikerét az a fejlesztési koncepció alapozhatta meg, amely szerint a csoportos vezérlésekkel a hagyományos vezérlések gazdaságos vagy perspektivikusan gazdaságos konkurrensait kell létrehozni. Megemlíthető a japán rendszerek teljessége: az alkatrészprogramozás, a gyártás, anyagmozgatás folyamatainak egységes rendszerben történő kiszolgálása.

A fenti példák rávilágítanak a csoportos vezérlések fejlődésének útjára: a gyártás folyamatainak integrálására egységes automatizált rendszerek keretében.

3.1.6 A számítógépes szerszámgépvezérlések kutatása Magyarországon

A számítógépes szerszámgépvezérlések kutatását Magyarországon az OMFB megbízásából az MTA SZTAKI-ban kezdték el, 1972-ben. A Csepeli Szerszámgépgyárral együttműködve folyó kutatás első célja a gyárban üzembehelyezendő csoportos vezérlés kifejlesztése.

Az MTA SZTAKI DNC'73 rendszere max. 8 szerszámgépet tud egyidejűleg, egymástól függetlenül vezérelni [100, 101, 102, 103]. Az egyes szerszámgépek mellett álló egységekben, MTC-kben /Machine Tool Controller/ igen korszerű kezelői interface, könegyenes interpolátor és adatátviteli egység található. A kezelői interface 12-karakteres alfanumerikus kijelzőt, számkijelzőt, jelzőlámpákat, számbeviteli és funkcionális tasztaturát tartalmaz. A gépek kezelése igen könnyen elsajátítható, előnyös dialógus formában történik.

A szerszámgépcsoport munkáját feladatlista alapján központi számítógép vezérli. A számítógép háttértárolója tartalmazza az alkatrészprogramok könyvtárát, ennek feltöltése, kezelése a diszpécser munkahelyéről lehetséges. Ugyansak a diszpécser munkahelyéről lehet az egyes gépek feladatlistáit módosítani, a termelés és a gépek pillanatnyi helyzetét lekérdezni, a rendszert elindítani és leállítani. Későbbi fejlesztéssel a diszpécser munkahely szolgáltatásai az alkatrészprogramok összeállításával, javításával lesznek bővíve.

A DNC'73 rendszer szerszám- és anyagmozgató berendezéseket is tartalmazhat, amelyeket hasonló rendszerben ugyanazon központi számítógép irányít.

Ha a DNC'73 rendszert a szerszámgépvezérlő rendszerek általános fejlődési irányai között kívánjuk elhelyezni, akkor megállapíthatjuk, hogy ez olyan csoportos vezérlés, amely csupán a közvetlen vezérlési funkciókat tekintve is gazdaságosan alkalmazható, szabad számítógépi kapacitása segítségével ugyanakkor több DNC funkciót is megvalósít. CAM rendszerbe való integrálása pedig természetes lehetőség, amelyet az alkatrészprogramozási interface különösen egyszerűvé tesz.

Az újabb fejlesztések közül megemlítjük az MTA SZTAKI Dialog CNC rendszerét [104]. Ez a rendszer 3D egyenes- 2D körinterpolációt végez, automatikus ekvidisztáns számítást tartalmaz. Kezelőpultja megegyezik a DNC'73 MTC egységeinek kezelőpultjával. A kezelőpulttól az alkatrészprogramok szerkesztése is elvégezhető. A vezérléshez programozható logikájú vezérlő készül [105].

3.2 Az NC berendezések vezérlési nyelvei

Ebben a pontban áttekintjük az NC berendezések azon funkcionális jellemzőit, amelyek a vezérlő nyelvben is tükröződnek.

A vezérlő nyelvek szabványosítására már a vezérlések széleskörű elterjedésével egyidőben jelentős erőfeszítések történtek. A kidolgozott szabványok és ajánlások /ISO R840, R841, R1057, R1058, R1059, R1056 [60, 122, 124] és származékaik/ széles körben elterjedtek, a vezérlések túlnyomó többsége alkalmazkodik a szabványokhoz.

A felmérés során nem támaszkodhatunk azonban kizárólag a szabványra, mert egyrészt a szabvány sok szabadságfokot hagy a tervezőnek, másrészt előfordulnak a szabványtól eltérő figyelemreméltó megoldások.

A lényegét jobban feltáró elemzéshez jutunk, ha a nyelveket a programozott funkciók szerint tekintjük át.

A vezérlések közvetlen nyelvén kívül felmérésünkben utalásokat teszünk egy másik szabványos nyelvre, az NC processzor outputjának használt CLDATA formátumra [61, 62, 125]. Tárgyunk szempontjából a CLDATA formátum jelentőségét az adja, hogy az információs folyamatban a vezérlési nyelvhez közelálló helyet foglal el: a két formátumot csak a posztprocesszor választja el.

Egyes funkciók jobb megvilágítása céljából a fejezetben helyenként utalunk arra is, hogy a magasabb szintű alkatrészprogramozási nyelvek az adott funkciót hogyan engedik megadni.

3.2.1 Mozgás-kontúrok programozása

Az egyenesszakaszokból és körivekből felépíthető kontúrok programozása az NC vezérlésekben meglehetősen jól szabványosított, a különböző vezérlésekben gyakorlatilag egyformán történik. A módszer lényege, hogy az alkatrészprogramban megadják két szomszédos pályaelem /egyenesszakasz vagy körív/ metszéspontjának /a pálya alappontjának/ koordinátáit és a két szomszédos alappontot összekötő pályaelem típusát. A körívek esetében a körüljárás irányát és a középpont koordinátáit /az érintő iránytangensét az egyik alappontban/ is meg kell adni.

A körívek ilyen megadása tulhatározottságot és redundanciát tartalmaz. [108] leír egy alternatív megadási módot, amely a vezérlések pályaszámítási egységeinek átalakulásával elterjedhet.

Érdekes a Siemens Sinumerik Sprint T vezérlésben alkalmazott konturleírási módszer, amely a szabványosnál jóval tömörebb.

A magasabb szintű geometriai nyelvek a konturmegadást nagymértékben leegyszerűsítik a technológus számára. Az itt alkalmazott módszerek implementálása növelné a kézi uton programozott vezérlések használati értékét.

A mozgás-pálya programozásával kapcsolatos a pályabejárás sebességének megadása. Az újabb rendszerekben általánossá vált az előtolási sebesség közvetlen megadása /mm/perc-ben/ a korábban szokásos módok helyett, és ugyancsak gyakori lehetőség a mm/fordulat megadási mód is. Különlegesség az olasz ECS cég 2100 sorozatu vezérléseinek az a megoldása, hogy a szükséges előtolási sebességet az elérendő felületi simaságból számítja ki. Ugyancsak ezt a vezérlést lehet példaképp felhozni arra, hogy esztergáknál a vágósebesség állandósítását az esztergálandó átmérő függvényében is automatizálni lehet.

3.2.2 Szerszám-korrekciók

A vezérlés és a programozó közötti munkamegosztást a mozgáspályák leírásának tekintetében meghatározza a programozandó referenciapont. Három eset különböztethető meg: a programozandó pont a szán referenciapontja, a szerszám egy referenciapontja vagy a mindenkor megmunkálási pont.[¶] Az első két esetet egyszerűsége miatt nem tárgyaljuk, csupán megjegyezzük, hogy a második esetben szükséges szerszám-hossz értékeket a vezérlés kívülről beállítható, programból hivatkozható memória rekeszei tartalmazzák /régebbi vezérlésekben erre a célra peremkereket alkalmaztak/.

[¶]A megmunkálási pont /a megmunkálás adott pillanatában/ a szerszám /vagy szerszámforgási burkolótest/ és készdarab /céldarab/ érintkezési pontja. A definícióval kapcsolatos a megközelítési probléma és a megmunkálási pont unikalitásának problémája. A megközelítési problémáról még történik említés, az unikalitás kérdése főleg 3D megmunkálásoknál merül fel, ehelyütt nem tudunk vele foglalkozni.

A megmunkálási pont programozása pályamenti mozgások esetén számottevő terhet ró a vezérlésre, mivel ekkor megmunkálási pontként a szerszám különböző pontjai léphetnek fel. Egyenes szakaszokból és körivekből, mint pályaelemekből felépülő síkbeli pályák körkeresztmetszetű /vagy köralakban lekerekített/ szerszámmal való megmunkálásánál ekvidisztáns pályát kell számítani. A vezérlésekben csak ennek az esetnek az automatizálása fordul elő.

Megjegyezzük, hogy külső sarkoknál technológiai okokból az ekvidisztáns pálya gyakran nem alkalmazható.

Az ekvidisztáns számítás algoritmikus bonyolultságáról megállapíthatjuk, hogy egymáshoz érintőlegesen csatlakozó pályaelemek esetén az ekvidisztáns pálya alappontjainak koordinátái a megmunkálási pont alappontjainak koordinátáitól lineárisan függenek, egyéb esetekben a függés másodfokú /négyzetgyökös/. Az említett függés együtthatóit legtöbbször a két szomszédos alappont ismeretében meg lehet határozni, de előfordulhat /pl. nagyoló megmunkálásoknál/, hogy több alappont ismeretére van szükség. Egyes esetekben az együtthatók numerikusan rosszul kondicionáltak.

A fejlődés mai fokán korszerűnek számítanak azok a vezérlések, amelyek az ekvidisztáns pálya alappontjait a már említett lineáris függés alapján kiszámítják. Nem érintőlegesen csatlakozó pályaelemek esetén ilyen algoritmussal csak szerszámsugár-kopáskorrekció végezhető el. Az együtthatókat általában az alkatrészprogramok tartalmazzák, de egyes vezérlések /köztük a magyar DIALÓG-CNC is/ az együtthatók kiszámítását is elvégzik. A vezérlések leírásából rendszerint nem állapítható meg, de valószínűsíthető, hogy az együtthatók több szomszédos alapponttól való függésének esete nem megoldott. Az egyszerű lineáris korrekciót azonban egy korszerű vezérlésnek mindenképpen tartalmaznia kell.

Megemlítjük a kontur megközelítésének problémáját. A megközelítési feladatot a vezérlések igyekeznek az ekvidisztáns számítási feladatra visszavezetni, mivel azonban a két feladat kissé eltérő, partikuláris, nehezen azonosítható megoldások születtek.

Végezetül kitérünk arra, hogy a magasabb szintű nyelvek processzorai az ekvidisztáns számítást hogyan automatizálják. Az általunk ismert processzorok közös vonása, hogy adott méretű szerszám referenciapontjának pályáját számítják, a megmunkálási pont pályája alapján. Az APT-III.

processzor például ezt a számítást három szabadságfoku maróra végzi el, a marószerszám alakja elég tág határok között változhat [114, 115].

A magasabb szintű alkatrészprogramozási nyelvek a megközelítési mód és a mozgás határolásának leírására jóval tágabb és könnyebben kezelhető lehetőségeket biztosítanak a programozó számára /lásd GO, FROM, IDIR utasítások megközelítéshez, ellenőrző felületek a mozgás behatárolásához/.

Az általános 3D probléma a 2D ekvidisztáns számítási problémára nem vezethető vissza. Alakos marók valamely referenciapontjának pályája az általános esetben például nem is ekvidisztáns a megmunkálási pont pályájával.

3.2.3 Sebesség kompenzáció

Az ekvidisztáns pályakorrekcióval kapcsolatban felmerül a pályamenti sebesség korrekciójának kérdése. A probléma lényege, hogy ha a megmunkálási pont görbült pályán mozog és a vezérlés szerszám-sugár korrekciót számít, akkor a megmunkálási pont sebessége és a szerszám középpont sebessége nem egyezik meg. Az irodalom alapján nem ismerünk olyan vezérlést, amely a sebesség szükséges korrekcióját automatikusan végezné.

3.2.4 Megmunkálási hibák korrigálása

A következő probléma a pálya korrekciójának problémája a szerszámgép geometriai pontatlanságának, a megmunkálás dinamikus viszonyainak megfelelően. Ilyen jellegű pontatlanságok miatt a megmunkálási pont pályája eltér a programozottól /a munkadarab koordinátarendszerében, amely koordinátarendszert a szerszámgéphez rögzítve képzelhetünk el/. A korrekció lényege, hogy a programozott pályát a hibával ellentétes értelemben korrigálják, ezáltal az eredő pálya az eredetileg kívánt pálya lesz.

Az egyszerűbb eset a szerszámgép statikus pontatlanságának korrekciója. A statikus hiba csak a szerszámgép szánjainak pillanatnyi helyzetétől függ.

A korrekciót egyszerűbb esetben csak a pálya alappontjaiban végzik el. Ezzel a korrekcióval növelik a szerszámgép pozicionálási pontosságát. Dinamikus hibakorrekciót valósít meg a svéd NUCON-400 CNC rendszer [55].

A szerszámgép statikus pontosságának korrekciójával azonos módon végezhető el az előtolómű statikus hibáinak kompenzációja /a vezérorsó, illetve a fogasléc menetemelkedési egyenetlenségeinek kompenzációja/ is. Hasonló jellegű feladat a holtjáték kompenzáció is, amelyet a szán irányváltásakor kell végrehajtani /pl. Sinumerik System 7, Bendix Dynapath/. Az irodalomban nem találtunk példát olyan vezérlésre, amely a holtjáték kompenzáció során figyelembe venné azt a tényt, hogy a holtjáték értéke a szán utja mentén más és más lehet.

Megemlítjük, hogy a kompenzációs értékeket egyes vezérlésekbe külön, speciális lyukszalagon kell bevinni, más vezérlésekben a kezelőpult nyomógombjairól végezhető el ez a művelet. Integrált gyártórendszerben szükség van arra, hogy a korrekciós értékek a mérési folyamatok eredményei alapján on-line csatornáról módosíthatók legyenek. Az irodalomból ilyen lehetőségek meglétére nem lehet következtetni.

Bonyolultabb probléma a megmunkálás dinamikus hibáinak kompenzációja. A dinamikus hibák egy részét számítás útján, már a megmunkálás tervezésekor meg lehet határozni, egy részüket méréssel lehet utólag azonosítani, a maradék összetevő a megmunkálás sztochasztikus hibája marad.

A dinamikus hibák kompenzációja a vezérlési programban megadott értékekkel kétféleképpen képzelhető el: vagy a mozgások sebességét korrigálják, vagy a pálya geometriáját. A sebesség korrigálásának esetében a vezérlésnek képesnek kell lennie az előtoló sebesség előre megadott függvény szerinti módosítására /idő- vagy helyfüggvény/. Programozható változatra nem ismerünk példát. Megemlíthetők azonban azok a vezérlések, amelyek nyelve megengedi a sebességkorrekció beszámításának engedélyezését /letiltását/ [118].

A pályageometria kompenzálására példa a Cincinatti Milacron Acramatic 10E vezérlése, amellyel az esztergált munkadarab hordósodásának, kuposodásának kompenzációja végezhető el. Véleményünk szerint hasonló jellegű szolgáltatások implementálása korszerű szerszámgépvezérlő rendszerben szükségszerű. Meg kell ugyanakkor jegyezni, hogy ezek kihasználása a technológia tervezésre, gyártás-irányításra speciális fela-

adatokat ró, ezért nem akadálymentes.

Kritikus megmunkálási feladat az éles sarkok kialakítása [59], mivel könnyen alávágás, illetve elévágás következhet be. Az ilyen jellegű hibák csökkentésére a vezérlések egyre hatásosabb megoldásokat alkalmaznak /növelik a rendszer határfrekvenciáját, stb./. Megjegyezzük, hogy a sarkok pontos kimunkálása egyes vezérlésekben többletidőt igényel, a mellékidők csökkentése érdekében ezért egyes vezérlésekben programozni lehet a pontos bejárás ki-bekapcsolását /Sinumerik 7M/. Az ilyen jellegű vezérlőnyelvi elemek azonban a vezérlések fejlődésével várhatóan elfognak tűnni.

3.2.5 Mozgáscsoportok programozása

A mozgáspálya programozás automatizálásának következő, egészen más jellegű módja az ismétlődő összetett mozgáscsoport rövid úton történő megadási lehetőségeinek bevezetése /alprogram, makrótechnika/. Számítógépek programozásában ez a technika alapvető és régóta ismert, az NC technikában az alkatrészprogramozási nyelvek szintjén azonban csak a legutóbbi időben terjedhetett el. E technika technológiai lehetőségeinek teljes kihasználása ezért még nem lehet teljes.

Az alprogramtechnikának különböző bonyolultságu megvalósításai lehetségesek, a már régóta ismert és szabványosított "fix ciklusoktól" kezdve [60, 61, 62] egészen a tipustechnológiák adott feladatra való lehívásáig. A bonyolultsági fokozatok abban különböznek egymástól, hogy a mozgáscsoport milyen módon paraméterezhető, vagyis mely technológiai, geometriai jellemzői változtathatók lehívásról lehívásra.

A paraméterek "bemásolását" elvégző vezérléssel megoldható, hogy ugyanaz a program "hasonló" alakú, de kissé eltérő ráhagyási alakzatokra legyen alkalmazható [56, 119].

A vezérlések "tudásával" arányosan ma már egyre bonyolultabb lehetőségeket biztosítanak alprogramok definiálására, használatára /Summit/Dana Bandit, Bridgeport-I CNC, Siemens Sinumerik System 7, Philips NC 660, GE Mark Century 1050 stb./. A szokásosnál sokkal "többet tudó" megoldást ismertet [120] és e dolgozat 4. fejezete.

Ezen utóbbi megoldásokban a vezérlés lényegében átveszi a pályaszintézis feladatának megoldását. Ez növeli a vezérlés technológiai rugalmasságát, jelentősen egyszerűsíti a kézi programozást, és a gépi programozást is segíti. Ilyen, magas szinten paraméterezhető alprogramok implementálása egy korszerű vezérlésben ezért kívánatosnak mondható.

Az alprogramokhoz technológiai szempontból hasonló technika a ciklusszervezés. Ciklusok szervezésével elérhető, hogy egy-egy programrészt, az un. ciklusmagot a vezérlés többször is végrehajtsa.

Nem vezérlő rendszer szolgáltatása, de előremutató az OKI MICROAPT-BC [121] ciklusszervezési lehetősége. A ciklusváltozó itt egyben geometriai elemek paraméterenként is hivatkozható.

Végezetül egy megjegyzés a magasabb szintű alkatrészprogramozási nyelvek által megvalósított alprogram- és ciklustechnikáról: E nyelvek lehetőségeit behatárolja a processzorok olyan felépítése, amely szerint a processzor először az összes geometriai elem definícióját kanonikus alakra hozza és csak azután kezdi meg a bejárési probléma megoldását. Ez a felépítés nem teszi lehetővé, hogy a geometriai leírást a mozgás eredményétől tegye függővé.

3.2.6 Feltételtől való függés programozása

A mai vezérlések többségének vezérlési programjai egyetlen típusú feltételes utasítást ismernek: a feltételes megállást. A megállás feltétele ebben az esetben egy kapcsoló állapota, amelyet a megmunkálás előtt állítanak két lehetséges helyzet valamelyikébe.

Megemlítjük a svájci ATEC cég MIPLAN alkatrészprogramozási nyelvét, amely egy program kétféle változatának megírását engedi meg egy programtörzson belül. Az éppen szükséges változat kiválasztása a megmunkálás előtt történik.

A feltételes utasításoknak ciklus szervezéséhez való felhasználására nem találtunk példát, holott a feltételtől való függés lehetősége növeli a vezérlési rendszer rugalmasságát. Feltételes utasítások implementálása ezért kívánatosnak mondható.

3.2.7 Kapcsolási funkciók

E dolgozat első fejezetében már említettük, hogy a vezérlő programok a szerszám-munkadarab relatív mozgását leíró utasításokon kívül a kapcsolási funkciók kiváltására szolgáló utasításokat is tartalmaznak. A kapcsolási funkciók tartalmi sokfélesége a szerszámgépek sokféleségéből következik. Az azonos funkciók azonos módon történő programozására a vezérlési nyelv vonatkozó szabványai [60, 122, 124] indítanak. A hagyományos vezérlésekben azonban ezt a célt nem sikerült elérni.

Más a helyzet a CLDATA formátummal [61, 62, 125]. Ebben az egyes kapcsolási funkciókat tartalmuk, hatásuk szerint programozzák, függetlenül attól, hogy az adott funkciót az illető gépen milyen különleges parancssorozattal kell kiváltani. A CLDATA formátum szerszámgép-független, hatás szerinti kapcsolási funkció programozása ezen funkciók programozásának legfejlettebb formája. Különleges probléma a szerszámgép-független formátumból a hagyományos vezérlési formátumba való konvertálás /az ún. posztprocesszálas/. Az eredményekről [26] ad átfogó képet. Megemlítjük [123]-at, ahol arról számolnak be, hogy a szerszámgépvezérlést közvetlenül CLDATA formátumu programmal irányítják.

Megállapíthatjuk, hogy a korszerű vezérlésben funkcióra orientált, következőképpen szerszámgép-független kapcsolás programozás biztosítása lenne kívánatos.

3.3 A számítógépes szerszámgépvezérlés programozásának követelményrendszere

A 3. fejezet eddigi részei alapján már kirajzolódnak a korszerű számítógépes szerszámgépvezérlés programozhatóságával szemben támasztható követelmények. Ezek konkrét megfogalmazása előtt azonban meg kell említeni R.Sim cikkét [154]. Az elképzelések fő irányának egyezésén kívül [101] ezt az is indokolja, hogy a meglevő rendszerek legértékesebb kritikáját éppen ez a munka tartalmazza.

R.Sim cikkében a DNC rendszer vezérlő minigépén futó NC processzor ideális output formátumát kívánja meghatározni. Ebből a célból az APT CLDATA és az ISO cserélhető, változó blokkformátum tulajdonságait értékeli.

R.Sim megállapítja, hogy a hagyományos feldolgozásnál fontos követelmények /a számbábrázolás feleljen meg a nagy gép aritmetikájának, maximális szerszámgép-függetlenség, jó hibadiagnosztikai rendszer kiépítésének lehetősége/ ma már háttérbe szorulnak a tömörség, azaz NC-hez közeli, kevés összfeldolgozást igénylő, könnyen szerkeszthető és olvasható formátum igényével szemben.

R.Sim követelményrendszeréhez néhány megjegyzést szükséges fűzni.

Először, mi az olvashatóság fogalmába nem értjük bele a közvetlen kiírhatóságot. A tömörség és a könnyű feldolgozhatóság érdekében szakítani kívánunk a karakterformátumon alapuló fizikai reprezentációval.

Másodszor, a többszörös feldolgozhatóság /más szerszámgépre való átirányíthatóság/ az irodalom szerint olyan cél, amelyet a gyakorlatban nagyon ritkán sikerült megvalósítani, nem elsősorban az átkódolási problémák miatt. Helyette ésszerűnek tartjuk a minimális szerszámgép-függetlenség követelményének kitűzését.

[154] és [101] gondolatmenetének továbbvitelével, a fenti megfontolások figyelembevételével és a 3. fejezetből kirajzolódó trendvonal alapján már meghatározhatók a korszerű vezérlés programozási nyelvvel kapcsolatos reális elvárások.

Ezek:

- optimális munkamegosztás a technológia tervezés /műveletelem tervezés/ és vezérlés között;
- minimális összfeldolgozás igényű alkatrészprogramozás-vezérlés biztosítása;
- szerszámgép-függetlenség gépcsoporton belül vagy adott megmunkálási módra;
- visszahivatkozások forrásnyelvi utasításokra a programszerkesztés követelményei szerint;
- program elemek egymásra hivatkozásának lehetősége a szerkeszthetőség, adaptálhatóság érdekében;
- determinisztikus programok helyett technológiailag megalapozott paraméterezhetőség;
- programozott operátori dialógus lehetősége;

- felkészülés többszános szimultán megmunkálásokra;
- kapcsolat a termelésirányítással;
- megmunkálási pontosság növelése program eszközökkel, adaptálhatóság a méretellenőrzés eredményei alapján;
- előre gyártott, magas szinten paraméterezhető "konzerv" technológiák alkalmazása;
- belső fizikai reprezentáció a számítógépes belső adatábrázolás követelményei szerint, a külső fizikai reprezentáció a technológus igényeihez igazítva. A kettő közötti transzformációk számítógép segítségével történő elvégzése.

A követelményekkel kapcsolatban az alábbi megjegyzéseket kell tenni.

Az első követelmény a 3.2.6 pont szerint szoros kapcsolatban áll az előre gyártott, magas szinten paraméterezhető "konzerv" technológiák alkalmazásának követelményével. A pályaszintézis egyes feladatait magára vállaló vezérlések könnyebben illeszthetők olyan művelettervező rendszerekhez mint pl. a TAUPROG-T. A pályaszintézist is elvégző technológiai tervező rendszer alkalmazása esetén /pl. FORTAP/ lehetővé válik a rendszer algoritmusának egyszerűsítése. Ez rendszertechnikai előnyt jelenthet. Technológiailag a vezérlés által végzett pályaszintézis dinamikus fogásmélység korrekciót és egyéb irányu program adaptálási képességet biztosít.

A technológiailag megalapozott paraméterezhetőség a vezérlő programok törzsére, tehát nem az alprogramokra vonatkozik. Hatása szintén a program adaptálhatóságában jelentkezik.

A termelésirányítással való kapcsolat nyelvi eszközei a program érvényességi körét, alkalmazhatósági feltételeit deklaráló utasítások és a paraméteres programozás eszközkészlete.

A számítógépes környezet lehetővé teszi a nyelv külső fizikai reprezentációjának /vagyis az input formátumnak/ a felhasználó igényeihez igazítását, esetleg különböző célokra különböző reprezentációk kidolgozását. A belső reprezentáció eközben változatlan maradhat, kialakítását a valós idejű feldolgozás, a vezérlések software-jének egységesítése hatá-

rozhatják meg.

3.4 Néhány vezérlőprogram formátum általános kritikája

Ebben a pontban a fenti követelmények fényében először kritikai szemmel megvizsgáljuk az ISO-EIA és az APT-CLDATA nyelvek főbb tulajdonságait, általános fejlődését, majd értékeljük a kapcsolódó egyéb irodalmat.

Az ISO-EIA és az APT-CLDATA formátummal kapcsolatban ismét idézzük R.Sim cikkét. Sim az APT-CLDATA formátumnál a következőket állapítja meg:

Az APT-CLDATA formátum teljes mértékben biztosítja a szerszámgéptől való függetlenséget, de nem annak minimális szintjén. A vezérlések tulajdonságaiban többsége ugyanis nagyjából hasonló, szabványosított bemenő formátumot használ, ezért a kimenő formátumot a szerszámgép-függetlenség számottevő csorbitása nélkül közelebb lehetne hozni a vezérlések bemenő nyelvéhez. A CLDATA formátum ezenkívül a csak a vezérlést ismerő szakember számára nehezen olvasható, javítása, módosítása nehézkes.

Az APT-CLDATA formátum értékelésével egyet lehet érteni. Hozzáteesszük, hogy a CLDATA-ban alkalmazott visszahivatkozási rendszer a forrásnyelvű program utasításaira egy geometriai processzor szemelőtt tartásával lett kidolgozva. Integrált geometriai-technológiai tervező rendszer esetén más visszahivatkozásokra lesz szükség. E tekintetben a FORTAP rendszer [156] közbenső nyelve [157] irányt mutató.

Az ISO/EIA szabványok szerinti formátumot a minigépen futó NC processzor kimenő formátumával szemben általa felállított követelményekkel összevetve Sim a következőket találja:

Az ISO /cserélhető/ változó blokkformátum nagyon közel áll a konkrét vezérlések bemenő formátumához. Hátránya, hogy nem szerszámgép-független. További problémát jelent, hogy nem minden posztprocesszor szót lehet ebben a formátumban megjeleníteni. A problémát a legújabb EIA ajánlásokban [118] is szereplő "control in"- "control out" zárójelek közé irt, a posztprocesszor vagy a /CNC/ vezérlés számára szóló utasításokkal lehet megkerülni. A kérdéses formátum ugyanakkor közvetlenül kiiratható és elolvasható, az APT-CLDATA formátumnál jóval tömörebb.

Az NC processzor outputjaként történő használata lényegesen csökkentené a processzor-posztprocesszor együttes adatfeldolgozási feladatát, mivel elmaradna a végeredménytől igen eltérő közbenső formátum bekódolásának és dekódolásának szükségessége. Továbbmenve, egyes korszerű CNC berendezések esetében a posztprocesszort szinte teljesen el lehetne hagyni, amely esetben a posztprocesszor ellenőrzési funkcióit egy külön, az alkatrészprogramok ellenőrzésére, kirajzolására szolgáló számítógépes program vehetné át.

Az ISO formátum hátránya, hogy használata esetén nehézkes a vezérlő programok többszörös feldolgozása, nevezetesen például egy adott programnak másik szerszámgépre történő átirányítása.

Az ISO formátum kritikájához hozzátesszük, hogy a programok szerkesztését nehezíti a koordináta értékekre is kiterjesztett modalitás. A modalitás miatt egy érték módosítása a pálya nem várt módosulásait is eredményezheti. Nem megoldott ugyanakkor az /a CLDATA formátumban sem/, hogy egy érték változtatásakor a programban több olyan pont helyzete változzon, amelyeket az adott ponttal nem modalitás köt össze.

Mindkét formátum Sim által nem elemzett hiányossága, hogy több olyan funkció megvalósítását, igény kielégítését nem teszik lehetővé, amelyekre gyártórendszer esetében szükség lenne. Ilyenek: a programok adaptálhatósága a gyártás körülményeihez, párhuzamos folyamatok programozása.

Emlitettük a FORTAP rendszer közbenső nyelvét. Ennek a nyelvnek a mondotton kívül több olyan vonása is van, amellyel egy, a korszerű számítógépes tervező-gyártórendszerekhez kialakított közbenső nyelvnek rendelkeznie kell. Ilyen vonás a nagyfokú hasonlóság a vezérlések alkatrészprogramozási nyelvéhez, a rekordstruktúra.

A FORTAP közbenső nyelv sajátossága, hogy a rekordokban közölt megmunkálási adatok nem szerszámgép-függetlenek. Hátrány, hogy a mozgási rekordok nem teszik lehetővé egyenesszakaszokkal interpolált konturok tömör leírását.

[160] szintén kapcsolódik e témakörhöz. Munka a vezérlési program redundanciáját veszi célba. Információelméleti alapon elindulva határozza meg a minimális redundanciával járó program tárolási formátumot /representációt/.

A tipustechnológiák tárolására alkalmas módszert ismertet [120]. A módszert számítógépes vezérlés nyelvére alkalmazva olyan rendszerhez lehet jutni, amelyben a vezérlési program a fogásosztás algoritmusát tartalmazza. Hibája, hogy az ismertetett módszer szerint nagyon nehéz közvetlen úton vezérlési programokat előállítani, a javaslat viszont nem tárgyalja a számítógépes módszerek lehetőségeit, problémáit.

[158] és vele párhuzamosan [159] két másik alternatívát ismertet:

az UNILEX és a TLDATA rendszer közbenső nyelvét. Mindkét interface kidolgozását az motiválta, hogy a szoborszerű felületek megmunkásához szükséges nagytömegű geometriai adatot a leghatékonyabban tudják reprezentálni. A rendszerek hiányossága az, hogy a kapcsolódó technológiai adatok reprezentációjára kevesebb figyelmet fordítottak.

Figyelemre méltó az utóbbi két rendszer előadása kapcsán a PROLAMAT'76 konferencián kibontakozott vita, amelyben W.H.P. Leslie konkrétan is kifejtette, hogy a meglevő rendszerek elterjedtsége az új rendszerekkel szemben milyen erős korlátozó tényező. A vitában ugyanakkor azt /a szerző véleményével egyező/ álláspontot is megerősítették, hogy a további fejlődés útjának keresését csupán ez a tény nem akadályozhatja meg.

4. A számítógépes szerszám gépvezérlés egy új programozási nyelve: a MID nyelv

Ebben a fejezetben röviden összefoglaljuk azokat a fontosabb jellemzőket, amelyek a szerző által kidolgozott MID nyelvet a többi vezérlési nyelvtől megkülönböztetik. A nyelv részletes specifikációja a függelékben található. Az elnevezés a nyelvnek a technológiai tervezési-vezérlési információfeldolgozási folyamatban betöltött szerepére utal.

Rendeltetés

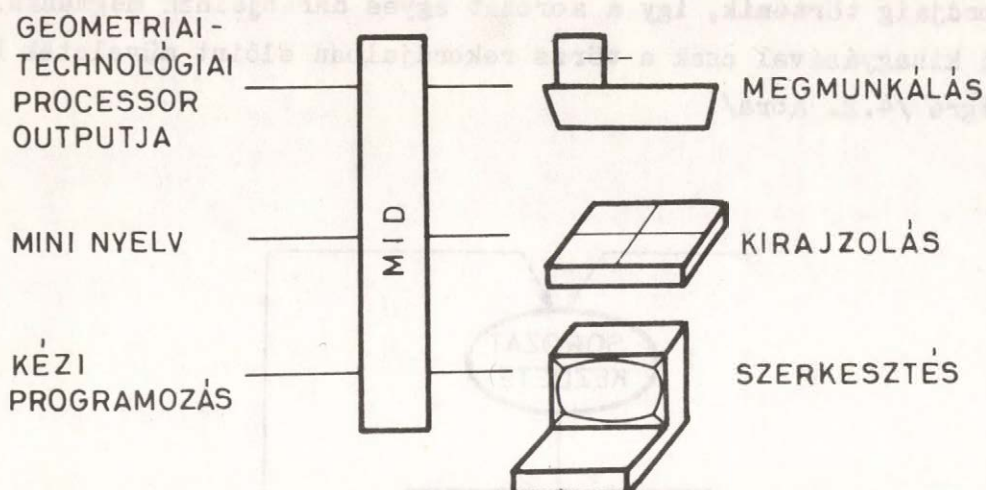
A MID nyelv a számítógépes szerszám gépvezérlések számára kidolgozott vezérlő nyelv. A MID nyelven kiválóan programozhatók egyedi CNC vagy csoportos, gyártórendszer jellegű vezérlések. Jelenlegi formájában a nyelv 2 1/2 D pályamenti megmunkálások programozására alkalmas, beleértve a megnövelt pontosságú megmunkálások esetét is. A nyelv kidolgozásánál az esztergálás, marás és furás műveleteit tartottuk szem előtt.

A MID nyelvű programot a gyártástervezés állítja elő, számítógépes segédlettel /4.1. sz. ábra/. Ezen utóbbi rendszer programozási nyelve lehet alacsonyszintű /ISO, EIA szabvány szerinti/, lehet valamilyen /párbeszédés/ közepes szintű nyelv /pl. MINIAPT/ vagy magasszintű geometriai vagy geometriai-technológiai nyelv. A számítógépes szerszám gépvezérlés a vezérlési programokat minden esetben MID formában tárolja.

A MID programok feldolgozása szintén számítógép segítségével történik. A leglényegesebb feldolgozás maga a megmunkálás, amelyet a számítógépes szerszám gépvezérlés vezérlő programja irányít. Egyéb feldolgozásokra is szükség van /pl. program ellenőrzésre, megjelenítésre, szerkesztésre stb./ ezek szintén számítógépes segítséggel történnek.

Fejrész-törzs

Egy munkadarab új sorozatának megkezdése előtt szükség van a munka felteteleinek ellenőrzésére: az alkatrész, a program, a nyersdarab, a gép-állapot, stb. azonosítására. Az azonosítás történhet automatikusan, vagy a szerszám gép kezelőjével folytatott dialógus segítségével. A lényeg az, hogy erre a műveletre minden sorozat legyártása előtt csak egyszer van szükség.



4.1. ábra: A MID nyelv helye a technológiai tervezés-gyártás rendszerében

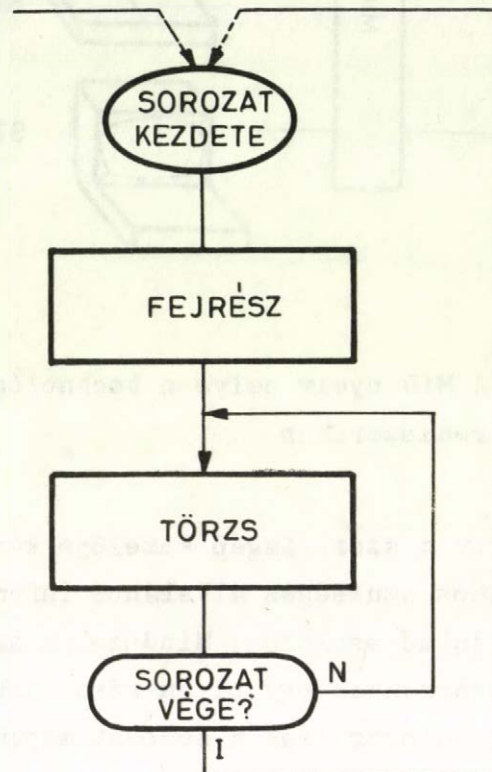
Uj sorozat elkezdésekor a szerszámgép kezelője kérheti, hogy az alkatrészprogram futtatásához szükséges általános információ jelenjen meg valamely megfelelő kijelző eszközön. Mindezeket az információkat célszerűen az alkatrészprogramnak egy olyan rész tárolhatja, amelyet a szerszámgépvezérlési rendszer csak a sorozat megmunkálásának kezdetén értelmezne.

Előfordulhat az az eset is, hogy egy munkadarabnak egymástól kismértékben különböző változatait egy alkatrészprogrammal lehet megmunkálni. Ebben az esetben a változat kijelölése a termelésirányítás feladata. Olyan megoldás is elképzelhető, amelyben a kezelő számára is nyitva hagynak néhány lehetőséget. Az ezek közötti választást vezérlő utasításokat az alkatrészprogram speciális részében kell elhelyezni.

Ezen feladatok megoldására a MID programokat két részre osztottuk: fej-
részre és törzsre.

A szerszámgép vezérlés új sorozat kezdésekor végigveszi a program elején elhelyezett fej rekordjait és az azokban található információk alapján elvégzi a szükséges azonosítást, inicializálást, stb. A széria további darabjainak megmunkálása előtt "visszacsvélés" csak a törzs

első rekordjáig történik, így a sorozat egyes darabjainak megmunkálása-
kor a fej kihagyásával csak a törzs rekordjaiban előírt műveletek haj-
tódnak végre /4.2. ábra/.



4.2. ábra: Fejrész és törzs végrehajtásának
algorithmusa

Az alkatrészprogram memória elemei - a MID változók

A MID nyelvű vezérlő programokban változó tartalmu memória elemek hasz-
nálatának lehetősége biztosított. Ezeket a memória elemeket MID válto-
zóknak neveztük.

A MID változók képezik az alkatrészprogramok adaptálhatóságának kul-
csát. Az alkatrészprogram végrehajtásának alábbi szituációiban jutnak

szerephez:

- a/ A vezérlő program módosítása a kezelő által közölt adatok alapján. Az irányítási rendszer a vezérlő programtól indítva az abban megadott /szöveges/ kérdést teszi fel a kezelőnek, a kezelő válaszát pedig a vezérlő programhoz rendelt adott memória elemében, a megfelelő MID változóban tárolja /pl. legyen-e előnagylás, mennyi a darab átmérője, stb./. A változóra vezérlő program megfelelő helyén hivatkozás történik.
- b/ A vezérlő programban megadott utasításra az irányítási rendszer a megmunkálás bizonyos paramétereinek más alrendszerek által számított értékeit kívánja átvenni, azért, hogy azokat a vezérlő programból is elérhető MID változóba tegye /pl. az előkészítő, mérő munkahely által megállapított tényleges méreteket, a keménység, anyagminőség globális becslését tartalmazó kódszámot, stb./.
- c/ A szerszámgép egyes állapotváltozásai, a kezelő lényegesebb beavatkozásai /pl. adatkérés, a ciklus start gomb megnyomása, vagy bármilyen jelzés, a programból kezdeményezve/ a vezérlő program végrehajtásának menetét MID változókon keresztül befolyásolják.

A MID változókra a vezérlő programból hivatkozni lehet, értéküket feltételes elágazási, feltételes szubrutinhívási utasítások végrehajtásához, feltételes értékadáshoz, elmozdulásérték, technológiai adat kiszámításához lehet felhasználni.

A MID változók értéküket tehát a megmunkálás során veszik fel és ezért azok csak a sorozat megmunkálásának idejére őrződnek meg.

/A MID változókról ld: Fl.1, Fl.2.2, Fl.2.3.4, Fl.2.3.6, Fl.2.3.7, Fl.2.3.8, Fl.2.4.2 pontokat/.

Rekordstruktúra

A számítógépes feldolgozás egyszerűsítése érdekében a MID mondatai /rekordjai/ explicite hordozzák típusuk kódját. A típus a rekordban található információ tartalmára, rendeltetésére és a rekord felépítésére utal.

Különlegesség, hogy a rekordok feldolgozásának menete programozható. A programozó előírhatja, hogy az adott rekord végrehajtásának milyen feltételt szab. A feltétel lehet a gép állapota, kezelő beavatkozása, kivárás stb.

/ld: Fl.1, Fl.2.1 példája/.

Az alábbi rekordtipusokat vezettük be:

- technológiai,
- az alkatrészprogram változóira vonatkozó,
- kommunikációs,
- a végrehajtást vezérlő,
- deklarációs

rekordok.

A technológiai rekordok a megmunkálás alapvető mozgáselemeit írják elő. A hagyományos szerszámgépvezérlések többnyire csak a MID nyelv technológiai típusu rekordjait hajtják végre.

A vezérlő programok változóira vonatkozó rekordok ezen változók értékadását végzik.

A kommunikációs rekordok egyrészt a szerszámgép kezelőjével folytatandó dialógusokat írják elő, másrészt a kapcsolódó rendszerekkel való kommunikációt szolgálják.

Ezekkel a rekordokkal lehet például üzenetet küldeni a szállítási rendszernek vagy a gyártásirányításnak, a program végrehajtásának előre meghatározott állapotának elérésekor.

A végrehajtást vezérlő rekordok egyrészt a végrehajtás sorrendjét vezérlik /vezérlésátadási utasítások/, másrészt a végrehajtás felfüggesztését, befejezését írják elő.

Ebbe a csoportba tartoznak az alprogramok végrehajtását kezdeményező utasítások is, továbbá párhuzamos folyamat indításával, leállításával kapcsolatos rekordok.

A deklarációs rekordok a vezérlő program azonosítására szolgáló, a megmunkálás feltételeire vonatkozó adatokat hordozzák. Ebben a csoportban tartozónak tekintjük a megmunkálási alprogramokat leíró rekordokat is.

Mozgási információ megadása

A MID nyelvben a megmunkálási pont pályáját kell programozni. A vezérlés feladata a szerszám sugárkorrekció /ekvidisztáns számítás/ elvégzése.

A run-time időszükséglet csökkentése érdekében az ekvidisztáns számítás paramétereit a MID program előprocesszált formában tárolja /ld.

Fl.2.3.3 pontban: "szerszámsugárkorrekció programozása"/.

Geometriai korrekció

A MID programban lehetőség van a pálya előre ismert, szisztematikus hibáinak pályatorzítás útján való kompenzálására. A megoldás különlegessége, hogy a vezérlés a kompenzációt a megmunkálási pont névleges pályájára szuperponálja. Ilymódon a geometriai korrekció és a névleges pálya külön kezelhető /ellenőrizhető, módosítható/. A geometriai korrekciőt a megmunkálási pont pályájához képest kell megadni. A vezérlés képes egyidejűleg szerszám sugárkorrekciőt és geometriai korrekciőt végezni. A run-time időigény csökkentése érdekében egy előprocesszor ebben az utóbbi esetben segédadatokat generál a MID programba.

Geometriai korrekcióval az alappontok koordinátái korrigálhatók, továbbá az egyenes szakaszok hordósodása kompenzálható. Ezzel a két lehetőséggel a pálya nagyon bonyolult alakváltozásai programozhatók /ld.

Fl.2.3.3 pontban/.

Kapcsolási információk

Funkcióra és eredményre orientált, szerszámgép-független módszert specifikáltunk. Különlegesség a szerszámtípus alapján történő hivatkozás. A megfelelő MID nyelvi rekordokat előfeldolgozás során generálják /Fl.2.3.5 pont/.

Szimultán folyamatok programozása

A MID nyelvben jelentős lépéseket tettünk az egymással szimultán folyó folyamatok programozása megvalósításának irányába. Ez a megoldás a feldolgozás számára jóval egyszerűbb, mint az ISO nyelv hasonló irányú kiterjesztései. A módszert a külső szerszámtár programozásának példájával illusztráltuk /Fl.2.3.6 pont/.

Paraméteres programozás

Nagy jelentőségű ujtás a mozgások és kapcsolási adatok paraméteres programozhatóságának bevezetése. A MID programban bármely számadat helyén MID változókra hivatkozó aritmetikai kifejezés állhat. A programot ezáltal rendkívül egyszerűen lehet a megmunkálás mért vagy megfigyelt jellemzői alapján módosítani. Programozási példát mutattunk be arra, hogyan teszi lehetővé a paraméteres programozás a nagyoló fogás módosítását a nyers darabnak a munkás által lemért adatai alapján. /Fl.2.3.7 pont/.

Alprogram technika

Rendkívül jelentős lépést tettünk előre gyártott, magas szinten parameterezhető "konzerv" technológiák tárolásának és feldolgozásának biztosítása érdekében. Az alprogram technika megvalósítása arra a felismerésre épül, hogy az ilyen un. műveletelem típusu alprogramok egyszerűbben írhatók egy olyan magas szintű nyelven, amelybe a hagyományos vezérlő nyelvet beleágyaszták. Definíáltuk ezt az un. gazda nyelvet és a beágyazás módját /Fl.2.4.3 pont/.

Minta /pattern/ típusu alprogramok bevezetésével sikerült megoldani furási pontmintázatok alprogramjaival kapcsolatos azon problémát, hogy a pontmintát leíró alprogram egy előre nem ismert, tehát paraméterként megadandó másik alprogramot hívjon /nevezetesen a furási ciklus alprogramját/. A paraméter listának a furási alprogram azonosítóján kívül természetesen annak paramétereit is tartalmaznia kell. A minta típusu alprogram bevezetésével a problémát a nyelv minimális bővítésével sikerült megoldani.

A műveletelem típusu szubrutinok az alábbi előnyöket biztosítják.

Először: lehetővé válik, hogy nemcsak az olyan technológiai paramétereket, mint a vágósebesség, előtolási sebesség, hanem a műveletelemben alkalmazott fogásmélységet is módosítsák, akár alkatrészről alkatrészre, a gyártás konkrét körülményeinek megfelelően /a nyersdarab mérete, a szerzőszám, a gép állapota szerint/.

Másodszor, a programokban olyan szubrutinok hívhatók, amelyek a műveletem geometriai és technológiai adatai alapján - kész mozgási sémák adaptálásával - a teljes pályaszintézist elvégzik. A műveletem geometriáját csak a szubrutin algoritmus korlátozza. Ez a technika rendkívül előnyösen használható akkor, ha gyakran kell hasonló, kismértékben különböző műveletelemeket tervezni. A hasonlóság kifejezést itt abban az értelemben használjuk, amelyet a szubrutin algoritmus meghatároz. A megfelelő geometriai hasonlóság-fogalom ebből származtatható.

A műveletem típusu szubrutinok alkalmazása esetén a programozói munka nagymértékben leegyszerűsödik, csökken a hibák elkövetésének valószínűsége. A programozónak csak azt kell megvizsgálnia, hogy az adott műveletelemre létezik-e a rendszer könyvtárában szubrutin, nem merülnek-e fel ütközési, hozzáférési problémák. Marásnál és nagyoló esztergáláskor a programozónak a bejárás stratégiáját is meg kell adnia /cikk-cakk, meander stb./. A műveletem típusu szubrutinok nagymértékben növelik a tipustechnológiák alkalmazásának gazdaságosságát.

Függelékben programozási példát adunk egy furási pontmintázat programozására és egy speciális, magas szintű pályaelem, a kvadrátikus b-spline egyenes szakaszokkal való közelítésére.

Fl. Függelék: A MID-szerszámgép vezérlési nyelv - részletes specifikációja

Az alábbi fejezet a számítógépes szerszámgépvezérlésnek a szerző által kidolgozott nyelvét a MID /közbenső/ nyelvet ismerteti. A nyelv elnevezése a MID programoknak a CAD/CAM rendszerben betöltött szerepére utal.

A nyelv teljes megismeréséhez a logikai formátum, vagyis a vertikális struktúra és a horizontális struktúra, valamint a fizikai formátum tárgyalására van szükség. Ezek közül a fizikai formátum erősen függ az adott hordozótól és mivel a logikai formátumhoz képest elvileg semmi újat nem tartalmaz, ismertetésétől helyütt eltekintünk.

A leírásban először a vertikális struktúrára jellemző általános szabályokat adjuk meg, majd a horizontális logikai formátumot ismertetjük. Ekkor teszünk említést a vertikális struktúrára vonatkozó olyan elemekről, amelyek az előző részből kimaradtak /pld. a modalitásról/.

Fl.1 A MID file vertikális struktúrája

MID file-nak nevezzük valamely gépészeti alkatrész gyártási folyamatát leíró programot, ha az az adott hordozón az alább specifikált formában van jelen.

A MID file logikai rekordok sorozatából áll, amelyek mindegyike egy logikai egységet képező információt tartalmaz.

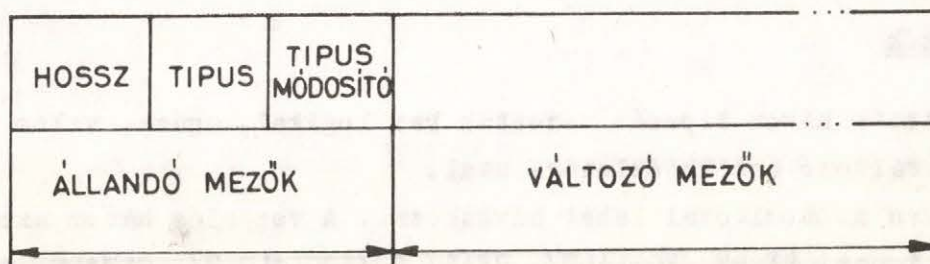
A logikai rekord alkotórészeit mezőknek nevezzük.

Állandó mezők /Fl.1 ábra/

Minden rekord első mezője a rekord hosszát tartalmazza, adathordozótól függő egységben: byteokban, karakterekben vagy szavakban.

A második mező a rekord típusát és a végrehajtást befolyásoló típusmódosítót tartalmazza. A rekord típusa a rekord tartalmára és felépítésére utaló jelzés. A típus módosító alapján dönthető el, hogy a rekordban előírt műveletcsoport megkezdése, vagyis a rekord értelmezése előtt milyen feltétel teljesülését kell bevárni. Megadható azonnali /feltétel nélküli/ vagy feltételes végrehajtás. A feltételes végrehajtás /kivárás/

a szerszámgép készenléti állapotához vagy egyéb eseményhez, mint feltételhez, köthető. Egyéb esemény előírása esetén a rekord következő mezője ezt a feltételt adja meg, a szükséges paraméterekkel együtt. Kivárási feltételt a törzs rekordjai tartalmazhatnak /ld. még az Fl.2.1 pont példáját/.



Fl.1 ábra: MID rekord felépítése

A MID file feldolgozását végző program sorra olvassa az egyes logikai rekordokat és megteszi a megfelelő lépéseket.

Minden új rekord feldolgozása akkor kezdődik, amikor az előző rekord értelmezésével kapcsolatos valamennyi tevékenység befejeződött, és a végrehajtás folytatásának a rekord típusa által meghatározott feltétele teljesül.

Ha tehát egy rekord több különböző technológiai művelet utasítását tartalmazza, akkor az azok végrehajtását kezdeményező parancsok egymástól függetlenül kimennek. Ezek a műveletek ily módon időben egyszerre zajlanak.

Változó mezők

A harmadik mező utáni mezők a rekord típusától függően változó tartalmúak. A mezők felépítése kétféle lehet. Állhatnak adatazonosítókból és a hozzájuk tartozó esetleges adatokból. Olyan rekordoknál, amelyeknél a mezők tartalmát a rekord típusa egyértelműen meghatározza, a mezők csak adatot tartalmaznak.

A mezők tartalmát az egyes rekordtípusoknál ismertetjük.

Fejrész, törzs

A MID file elején található rekordok a fejrész rekordjai. A fejrészt a "MID fej vége" /08 típusu/ rekord zárja. A fejrészben előforduló rekordtípusokat és a sorrendi megkötéseket az Fl.2.2 pont tartalmazza. A többi rekord a törzs rekordja. A MID programot "MID file vége" rekord zárja.

MID változók

A MID változók három típusát vezettük be: logikai, egész, valós típust. A típus a változó értékkészletére utal.

A változókra azonosítóval lehet hivatkozni. A vezérlés három azonosítót eleve ismer. READY /FOGLALT/, READY 2 /FOGLALT 2/, CSTART. A három azonosító rendre a szerszámgép főfolyamatáról, a mellékfolyamatról tudósító készenléti jel, illetve a kezelőpult ciklus start gombja. Mindhárom változó logikai típusu.

A fenti három MID változón kívül minden, a programban szereplő MID változót a fejrészben deklarálni kell.

Fl.2 A MID nyelv horizontális strukturája

Ebben a részben adathordozótól illetve számítógéptől függetlenül, részletesen ismertetjük a MID-rekordok logikai felépítését. /A rekordok fizikai felépítését minden adathordozóra és minden számítógép-belsőre külön meg kell adni/.

A leírásban sorra vesszük a MID formátumban specifikált rekordtípusokat. Először azok a rekordok következnek, amelyek csak a fejben fordulhatnak elő, vagy tipikusan a fejben alkalmazhatók. A többi rekord ismertetése a törzs rekordjainál történik.

F1.2.1 Jelölések

A rekordok felépítését a Backus-Naur féle metanyelv módosított verziójával írjuk le. Az egyszerűség kedvéért, ha ez nem vezethet kétértelműséghez, a metanyelvi egyenlőségek baloldalát elhagyjuk. A rekordok felépítésénél a hosszat, a típust, a típusmódosítót és a kivárási feltételt nem említjük.

A metanyelv kibővítése a $\left\{ \begin{smallmatrix} a \\ b \end{smallmatrix} \right\}$ és a $\left[a \right]_i^j$

jelölések bevezetéséből áll. A $\left\{ \begin{smallmatrix} a \\ b \end{smallmatrix} \right\}$ jelölés azt mutatja, hogy az adott helyen a, b elemek közül egy áll.

$\left[a \right]_i^j$ arra utal, hogy a legalább i-szer, legfeljebb j-szer ismétlődik.

Nem definiáljuk az alábbi elemeket:

<azonosító>

<kód>

<egész>

<valós>

<logikai>

<karakter string>

Néhány gyakran előforduló elem definíciója:

< MID változó > :: = <azonosító>

< MID címke > :: = < kód >

< szubrutin > :: = <azonosító>

Egy MID rekord az eléje írt 01-es rekord adatával címkézhető.

Példa rekord metanyelvi leírására:

Kapcsolási funkciók programozása a Fl.2.34 pont szerint 02 típusu rekorddal történik, amelynek felépítését a szövegben az alábbi kifejezéssel adjuk meg:

$$\left[\langle \text{adataazonosító} \rangle \quad \langle \text{adat} \rangle \right]_0^n$$

$$\langle \text{adat} \rangle ::= \langle \text{kód} \rangle \mid \langle \text{valós} \rangle$$

$$\langle \text{adataazonosító} \rangle ::= S \mid T \mid H \mid EL \mid TX \mid NM \mid SK \mid V$$

A baloldala nélkül közölt kifejezés teljes alakja:

$$\langle \text{kapcsolási rekord} \rangle ::= \langle \text{hossz} \rangle \text{ 02 } \langle \text{típus módosító} \rangle \left[\langle \text{kivárási feltétel} \rangle \right]_0^1$$

$$\left[\langle \text{adataazonosító} \rangle \langle \text{adat} \rangle \right]_0^n$$

ahol:

$$\langle \text{hossz} \rangle ::= \langle \text{egész} \rangle$$

$$\langle \text{típus módosító} \rangle ::= FN \mid FT \mid E$$

$$\langle \text{kivárási feltétel} \rangle ::= \left\{ \begin{array}{l} DT \quad \langle \text{időtartam} \rangle \\ DV \quad \langle \text{MID változó} \rangle \\ DP \\ DS \end{array} \right\}$$

$$\langle \text{időtartam} \rangle ::= \langle \text{valós szám} \rangle$$

A típusmódosítókat azonosítóval kódoltuk. FN, FT, E rendre a feltétel nélküli megkezdést, a foglaltsági jel megszűnésének kivárást, illetve a kivárási feltétellel megadott egyéb feltételt jelenti.

DT adatazonosítóval adott időtartamig történő kivárást, DV adatazonosítóval a kijelölt /logikai típusu/ MID változó állapotának "1" be változását, DP adatazonosítóval azt a feltételt írjuk elő, hogy a kivárást a foglaltsági jel megszűnése után a párhuzamos folyamat befejeződéséig tartson. DS adatazonosítóval a vezérlés arra utasítható, hogy a rekord végrehajtásával a hajtás lemaradásának teljes megszűnését várja meg. Kivárási feltételt akkor kell megadni, ha a rekordtípus módosítója egyéb feltételre utal.

F1.2.2 A MID file fejrésze

40-es rekord

MID file azonosítója

Felépítése: < Azonosító >

A MID file első rekordja, kitétele kötelező.

41-es rekord

Az alkatrész azonosítására szolgál.

Felépítése: < Azonosító >

41-es rekord kitétele kötelező, a MID file második rekordja.

42-es rekord

A vezérlő program változata

Felépítése:

< változatszám >

< változatszám > :: = < kód >

42-es rekord kitétele nem kötelező ha szerepel, a 41-es rekord után következik.

43-as rekord

Rajzszám

Felépítése: < rajzszám >

< rajzszám > :: = < kód >

Megadása kötelező.

44-es rekord

A szerszámgép azonosítója

Felépítése: < Azonosító >

A 44-es rekord kitétele kötelező, a 43-as után következik.

A 44-es rekord után feltétlenül következik egy 45-ös rekord. Egy MID file-ban több 44-45-ös rekord csoport is előfordulhat.

45-ös rekord

Gépállapot azonosítója

Felépítése: < Azonosító >

A gépállapot az adott szerszámgép valamennyi olyan jellemzőjének megkivánt állapotát azonosítja, amely az alkatrész legyártásához szükséges. Pl.: szerszámozás, készülékezés stb.

A gépállapot azonosítása jelenleg a gépállapot kódjának a vezérlőberendezés kijelző egységén való megjelenítéssel, a kezelő válasza alapján történik. A kezelő a kód és a nála levő gyártási dokumentáció alapján tételesen ellenőrzi, hogy a gép felszereltsége megfelel-e az alkatrész legyártásának.

47-es rekord

Nyersdarab azonosítója.

A rekord felépítése: < kód >

A 47-es rekordban megadott kód alapján lehet eldönteni, hogy az adott alkatrészprogramban feltételezett nyersdarab rendelkezésre áll-e.

Kitétele nem kötelező.

20-as rekord

Kommentár I.

Felépítése: < karakter string >

A vezérlő program rendeltetését stb. leíró általános szöveg, amely azonosítás céljából pl. a diszpécser munkahelyen megjeleníthető. A fejben helyezhető el.

A rekord használata nem kötelező.

21-es rekord

Kommentár II.

Felépítése: < karakter string >

A vezérlő program utasításaihoz fűzött szöveges kommentár, amelyet újraprogramozáskor, tanulmányozáskor célszerű megjeleníteni.

A programban bárhol elhelyezhető.

37-es rekord

MID változónak értékadás kezelőpultról

A 37-es rekord hatására a rekordban levő üzenet kikerül a szerszámgép mellett elhelyezett helyi vezérlés kijelzőjére, majd a rendszer bevárja a kezelő választ.

A beadott értéket /igen, nem, számérték/ a rendszer a rekordban megadott változónak adja.

A rekord felépítése:

$\langle \text{üzenet} \rangle \langle \text{MID változó} \rangle$
 $\langle \text{üzenet} \rangle :: = \langle \text{karakter string} \rangle$

37-es rekordtípus esetén a további feldolgozás a kezelő válaszának beérkezéséig szünetel. A kezelő rendelkezésére álló válaszolási idő limitált.

Az üzenet jelenleg 12 karakterből állhat. A törzsben is előfordulhat.

38-as rekord

MID változónak értékadás külső, on-line rendszerből.

Felépítése: $\left[\langle \text{MID változó} \rangle \right]_1^n$

A rekord értelmezése során a vezérlés a felsorolt MID változóknak azokat az értékeket adja, amelyeket a külső rendszerek /pl. a termelésirányítás/ az alkatrész legyártásának inicializálásakor megadtak.

A törzsben is előfordulhat.

50-es rekord

Az adott MID file-ban szereplő, nem szabványos jelentésű MID változók azonosítói, típusonként. Ha a MID file-ban MID változó nem szerepel, 50-es rekordot nem kell megadni.

A MID változókra való első hivatkozás előtt a MID file fejében ki kell tenni.

Felépítése:

$\left[\langle \text{Tipus} \rangle \langle \text{darab-szám} \rangle \langle \text{Azonosító lista} \rangle \right]_1^3$

$\langle \text{Tipus} \rangle :: = \langle \text{Valós} \rangle \langle \text{Egész} \rangle \langle \text{Logikai} \rangle$

$\langle \text{Azonosító lista} \rangle :: = \left[\langle \text{MID változó} \rangle \right]_n^n$

51-es rekord

Az adott MID file-ban hívott technológiai szubrutinok darabszáma és azonosítóinak felsorolása. Ilyen rutinok hiánya esetén 51-es rekordot nem kell megadni.

A felsorolás azért szükséges, hogy a rutin elérhetőségét ellenőrizni tudjuk, és a szükséges inicializálásokat el tudjuk végezni.

A fej rekordja.

Felépítése:	$\langle \text{Darab-} \rangle \left[\langle \text{Azonosító} \rangle \right]$	Darabszám
	szám	Darabszám

52-es rekord

Az alkatrész legyártásához szükséges idő számított értéke, 0,01 perc-ben.

52-es rekord kitétele jelenleg nem kötelező.

08-as rekord

MID fej vége.

A MID file fejrészének végén kell kitenni.

Fl.2.3 A MID file törzse

Fl.2.3.1 Rekordok azonosítása

Azonosításra 01 típusu rekordok szolgálnak.

A rekord egy egész számot tartalmaz, amely megfelel a hagyományos vezérlésekben alkalmazott mondatszámoknak.

Ezt a mondatszámot a MID file szerkesztésénél stb. azonosításhoz lehet használni. A mondatszám ezenkívül az alkatrész legyártása során standard kijelzésként, más adatokkal együtt megjelenik a szerszámgépvezérlő egység kijelzőjén.

A 01 típusu rekordok a MID file rekordjai között bárhol előfordulhatnak.

F1.2.3.2 Mozgási utasítások megadása

Mozgási utasításokat 03 típusu rekordokban lehet megadni.

A rekord felépítése:

$$\left[\begin{array}{c} \langle \text{adataazonosító} \rangle \langle \text{adat} \rangle \\ 1 \end{array} \right]^n$$

$\langle \text{adataazonosító} \rangle :: =$ G | X | Y | Z | I | J | K |
CRX	CRY	CRZ	GK1	GK2	GK3	GK4
XM	YM	ZM	F	FR	FE	FP
CRI	CRJ	CRK				
KAX	KAY	KAZ				

F1.2.3.3 Mozgási rekordokban használatos adataazonosítók jelentése

Pályaelem típusa

G azonosító után a pályaelem típusát meghatározó kód következik. Az előírt típus újabb megadásig érvényes. A MID file első 03 típusu rekordjában a pályaelem típusát kötelező előírni.

A pályaelem típust definiáló G funkciók kódjai:

- G01 egyenes
- G02 kör, óramutató járásával egyező
- G03 kör, óramutató járásával ellentétes
- G33 csavarmenet

Alappontok koordinátái

A koordinátaérték megadásának módja:

$\langle \text{koordináta irány} \rangle \langle \text{koordináta} \rangle$
 $\langle \text{azonosítója} \rangle \langle \text{érték} \rangle$

$\langle \text{koordináta irány} \rangle ::= X | Y | Z$
 $\langle \text{azonosítója} \rangle$

$\langle \text{koordináta} \rangle ::= \langle \text{valós} \rangle$
 $\langle \text{érték} \rangle$

Az irányok azonosítására az ISO ajánlásnak megfelelően a konkrét szer-
számgeptől függően X, Y, Z stb. adatazonosítók használandók.

Az alappontok koordinátáit abszolút koordinátarendszerben kell megad-
ni, 0001 mm-es egységekben.

Az elmozdulások megadására szolgáló koordinátarendszert önmagával pár-
huzamosan akárhányszor el lehet tolni és az elmozdulásokat az ily módon
keletkezett rendszerben is meg lehet adni. /ld. 14-es rekord/.

A koordinátarendszer elforgatása jelenleg nem megengedett.

A vezérlési program minden újabb végrehajtásakor az első nullpontelto-
lásig egy alapkoordinátarendszer, nullponteltolástól újabb nullpontel-
tolásig az eltolt koordinátarendszer érvényes.

Az elmozdulások közül csak azokat kell megadni, amelyek irányában tény-
leges elmozdulás történik.

I, J, K adatok

I, J és K azonosítókkal körív interpolálása esetén a körív kezdőpontjá-
ból a körívhez huzott sugár hosszúságu érintőszakasz X, Y és Z irányu
vetületeit kell megadni.

Ha a vezérelt rendszer csak kétdimenziós interpolátort tartalmaz, és
így csak koordinátasíkban fekvő körívek interpolálására alkalmas, ak-
kor az I, J és K adatok közül mindig kettőt kell megadni, az aktuális
síknak megfelelően.

Előtolási sebesség megadása

F adatazonosító után a konstans pályamenti előtolási sebességet kell
megadni, mm/min, FR adatazonosító után mm/fordulat, FE adatazonosi-
tó után mm/fog egységekben.

F, FR, FE adatazonosítóval megadott előtolás értékek öröklődnek.

Gyorsmenet F, FR vagy FE adatazonosító után álló nulla értékkel programozandó. A gyorsmenet csak az adott mozgáselemre vonatkozik, az előző rekordra érvényes előtoló sebesség a következő, nem gyorsmenetű mozgáselemre automatikusan átöröklődik. Gyorsmenettel kör nem programozható.

A program első mozgáseleme előtt előtoló sebesség megadása kötelező.

Szerszámsugárkorrekció programozása

Mivel a MID-file a koordináta adatok között a kialakítandó kontur adatait kéri, ezért a feldolgozó program a MID file-ban megadott koordináta adatokból és a marószerszám sugarából illetve az esztergáló szerszám csucssugarának értékéből a szerszám referencia pont /középpont/ pályáját kiszámítja. Bizonyos esetekben ez a számítás igen munkaigényes, ezért a munka nehezét a MID-et előállító processzorra hárítjuk.

A MID-et feldolgozó program a szerszámsugár szerinti korrekciót a következőképpen végzi el:

A szerszám hosszának, a készülék méreteinek és a programozott nullponteltolásnak megfelelően korrigálja a pálya következő alappontjának programozott helyzetét.

Ha az aktuális pályaelem adatai között CRX, CRY vagy CRZ érték szerepel, akkor a vezérlés a szerszám adataiból kiolvasott R /sugár/ értékkel tovább korrigálja a következő alappont helyzetét. Körív típusu pályaelem esetén a körív középpontjának helyzetét megadó I, J, K adatokat is módosítja:

$$\begin{aligned} X' &= CRX \cdot R + X ; & I' &= CRI \cdot R + I_{MID} \\ Y' &= CRY \cdot R + Y ; & J' &= CRJ \cdot R + J_{MID} \\ Z' &= CRZ \cdot R + Z ; & K' &= CRK \cdot R + K_{MID} \end{aligned}$$

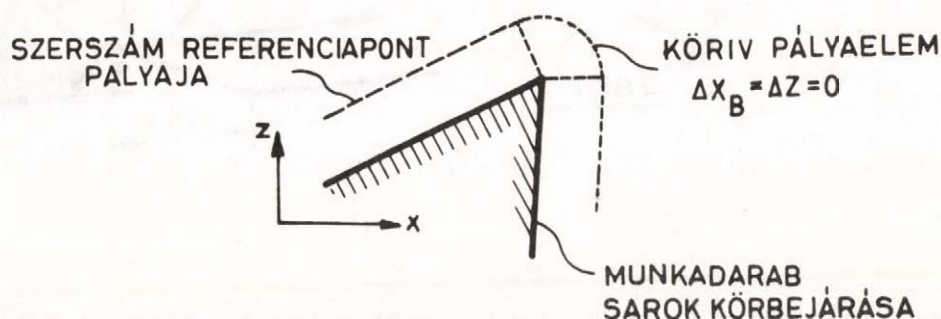
ahol X, Y, Z a megfelelő hosszkorrekciók beszámításával kapott, I_{MID} , J_{MID} , K_{MID} a programozott érték.

CRX, CRY, CRZ értékek mondatról mondatra nem öröklődnek. Nem megadott érték esetén a vezérlés nullát tételez fel.

Megjegyzés

Sarkok bejárásánál szükség lehet olyan pályaelemek /körívek/ programozására, amelyekben a ΔX , ΔY , ΔZ értékei mind nullák /a megmunkálási pont nem mozdul el, Fl.2. ábra/.

Az elmozdulást ilyenkor csak a CR értékek határozzák meg. A MID-ben ilyen pályaelemek megadása megengedett.



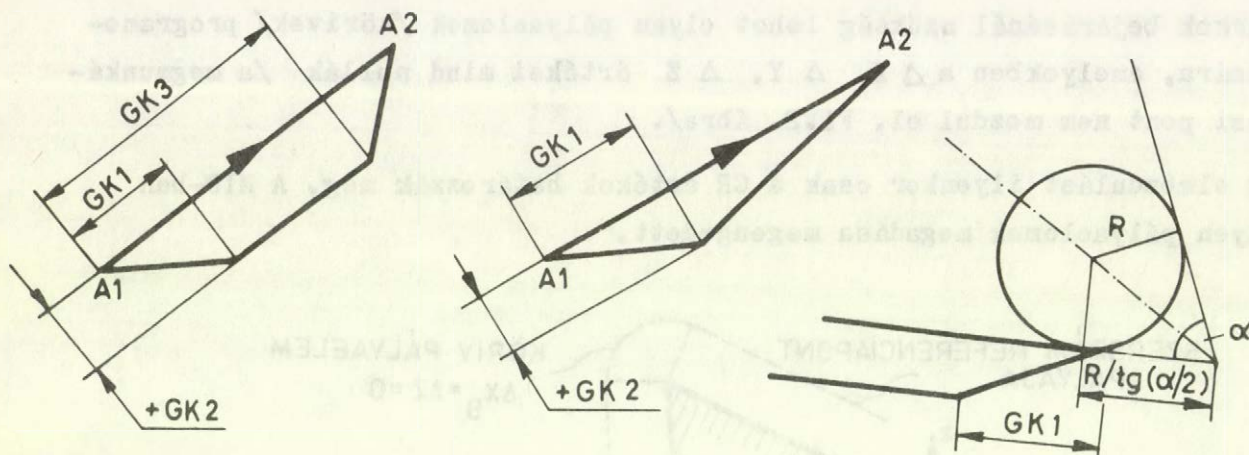
Fl.2 ábra: Sarok körbejárása

Geometriai korrekció

A geometriai korrekciót csak a 2D megmunkálás esetére dolgoztuk ki. A korrekció mindig a megmunkálás síkjában értendő.

Hordósodás kompenzáció GK1, GK2, és GK3 adatazonosítókkal oldható meg /Fl.3 ábra/. A GK adatok a korrekciót a pályaelem kiindulási alappontjához képest adják meg. A hordósodás kompenzációt a vezérlés a szerszámsugárral korrigált pályán végzi el, ezért a ténylegesen megvalósuló korrekció a szándékolt korrekciótól eltér. Az eltérés kompenzálása céljából GK4 adatazonosítóval megadható $\pm 1 / \operatorname{tg}(\alpha / 2)$, ahol α a pálya két szomszédos eleme által bezárt szög. Az előjel /-/ hegyesszög, /+/ tompaszög esetén. A vezérlés $GK1' = GK + R \pm GK4$ kifejezés szerint kompenzál, R a szerszám sugara.

Az alappontok korrekcióját KAX, KAY, KAZ adatazonosítók közül a megmunkálás síkjának megfelelő kettővel lehet előírni. Az alappontok korrekcióját a vezérlés a szerszámsugárkorrekció megkezdése előtt végzi el.



Fl.3. ábra: Hordosodás kompenzáció megadása három, illetve két paraméterrel
GK1 korrigálása a szerszámsugárral.

Előtolási sebesség korrekciója

FK adatazonosító után következő számérték az aktuális előtoló sebesség szorzótényezője. A vezérlés az aktuális sebességet korrigálja. Új előtoló sebesség beállítására a korábbi korrekció hatástalan.

Csavarmenet megmunkálása

Egybekezdésű menet: G33 kód esetén x, y, z a célkoordináták értékét adják /palást, sík és kupos menet lehetséges/, a menetemelkedés nagyságát a megfelelő I, J, K adatok tartalmazzák.

Pályabejárás pontossága befolyásának lehetőségei

FP adatazonosító egyirányú pozícióba érkezést ír elő. Negatív irányból való pozicionálás esetén a vezérlés automatikusan tulvezérel majd pozitív irányból pozícióba állít. Minden vezérelt tengelyre végrehajtódik. Nem öröklődő kód. Kör bejárására nem alkalmazható.

A kivárási feltételek között említett DS adatazonosító is a bejárasi pontosság befolyásolását szolgálja.

Programozás gépi koordinátarendszerben

XM, YM, ZM adatazonosítókkal megadott számértékek nem a megmunkálási pont, hanem a szán referenciapontjának helyzetét írják elő. Munkadarabcserénél, szerszámváltásnál használható. Kör menti bejárás nem programozható.

Fl.2.3.4 Kapcsolási funkciók programozása

Kapcsolási utasításokat 02 típusu rekordokban lehet programozni.

A rekord felépítése:

$$\left[\begin{array}{cc} \langle \text{adatazonosító} \rangle & \langle \text{adat} \rangle \end{array} \right] \begin{array}{c} n \\ 0 \end{array}$$

$\langle \text{adat} \rangle :: = \langle \text{kód} \rangle | \langle \text{valós} \rangle$

$\langle \text{adatazonosító} \rangle :: = S | T | H | EL | TX | NM | SK | V | TC | VK$

Fl.2.3.5 Kapcsolási rekordok adatazonosítóinak jelentése

Főorsó fordulatszám vezérlése: S adatazonosító után a fordulatszám értéke adandó meg. Kétirányú forgás lehetősége esetén a fordulatszám értéket a forgásirányra utaló jelzés követi /OJE, OJM/.

Főorsó leállítása: fordulatszám = 0.

Fokozatos főhajtás esetén a vezérlés a megadott értékhez legközelebb eső értéket kapcsolja.

V adatazonosítóval a vágósebesség értéke adható meg. A vezérlés a fordulatszámot az átmérő függvényében állítja be és szükség esetén /homlokesztergálásnál, pályamenti marásnál/ automatikusan korrigálja.

S, V értékek öröklődnek.

Főorsó fordulatszám korrekció. SK adatazonosító után következő számérték az aktuális fordulatszám szorzótényezője. A vezérlés az aktuális fordulatszámot korrigálja. Új fordulatszám megadásakor a korábbi korrekció hatástalanná válik. A vágósebességgel megadott fordulatszám VK adatazonosító után álló adattal korrigálható.

Pozíció alapján azonosított szerszám beváltása. T adatazonosítót a revolverfej pozíció kódja követi. Az utasítás hatására a vezérlés szerszámcserét hajt végre. A pozíció szerinti azonosítással szemben a szerszámtípus szerinti azonosítást érdemes előnyben részesíteni.

Tipus alapján azonosított szerszám kiválasztása, beváltása /04-es, 24-es típusu rekord/

Előváltás alatt a szerszámgép külső szerszámtartójának működtetését értjük, a beváltás egy szerszám megmunkálási helyzetbe hozása. A két művelet közül azt, amellyel a szerszámmal való tevékenység kezdődik, a konkrét szerszám kiválasztása előzi meg.

A kiválasztással együtt előváltást /TP/ vagy beváltást /TC/ is elő kell írni. A kiválasztást 04-es vagy 24-es rekordban kell programozni, attól függően, hogy a kapcsolódó előváltás illetve beváltás főfolyamatban vagy párhuzamos folyamatban zajlik. A már előváltott szerszám beváltása 02-es rekordban TC adatazonosítóval történik, mivel a beváltás kapcsolási funkció. A kiválasztáshoz szükséges adatok ilyenkor nem szerepelnek.

A lehetséges szekvenciákat és a megfelelő rekordtípusokat az Fl.4 ábra szemlélteti:

51-es rekord

Az adott MID file-ban hívott technológiai szubrutinok darabszáma és azonosítóinak felsorolása. Ilyen rutinok hiánya esetén 51-es rekordot nem kell megadni.

A felsorolás azért szükséges, hogy a rutin elérhetőségét ellenőrizni tudjuk, és a szükséges inicializálásokat el tudjuk végezni.

A fej rekordja.

Felépítése:	$\left\langle \begin{array}{c} \text{Darab-} \\ \text{szám} \end{array} \right\rangle \left[\left\langle \text{Azonosító} \right\rangle \right]$	Darabszám
		Darabszám

52-es rekord

Az alkatrész legyártásához szükséges idő számított értéke, 0,01 perc-ben.

52-es rekord kitétele jelenleg nem kötelező.

08-as rekord

MID fej vége.

A MID file fejrészének végén kell kitenni.

F1.2.3 A MID file törzse

F1.2.3.1 Rekordok azonosítása

Azonosításra 01 típusu rekordok szolgálnak.

A rekord egy egész számot tartalmaz, amely megfelel a hagyományos vezérlésekben alkalmazott mondatszámnak.

Ezt a mondatszámot a MID file szerkesztésénél stb. azonosításhoz lehet használni. A mondatszám ezenkívül az alkatrész legyártása során standard kijelzésként, más adatokkal együtt megjelenik a szerszámgépvezérlő egység kijelzőjén.

A 01 típusu rekordok a MID file rekordjai között bárhol előfordulhatnak.

F1.2.3.2 Mozgási utasítások megadása

Mozgási utasításokat 03 típusu rekordokban lehet megadni.

A rekord felépítése:

$$\left[\begin{array}{c} \langle \text{adataazonosító} \rangle \langle \text{adat} \rangle \\ 1 \end{array} \right]^n$$

$\langle \text{adataazonosító} \rangle :: =$ G | X | Y | Z | I | J | K |
CRX	CRY	CRZ	GK1	GK2	GK3	GK4
XM	YM	ZM	F	FR	FE	FP
CRI	CRJ	CRK				
KAX	KAY	KAZ				

F1.2.3.3 Mozgási rekordokban használatos adataazonosítók jelentése

Pályaelem tipusa

G azonosító után a pályaelem típusát meghatározó kód következik. Az előírt típus újabb megadásig érvényes. A MID file első 03 típusu rekordjában a pályaelem típusát kötelező előírni.

A pályaelem típust definiáló G funkciók kódjai:

G01 egyenes
G02 kör, óramutató járásával egyező
G03 kör, óramutató járásával ellentétes
G33 csavarmenet

Alappontok koordinátái

A koordinátaérték megadásának módja:

$\left(\begin{array}{c} \text{koordináta irány} \\ \text{azonosítója} \end{array} \right) \left(\begin{array}{c} \text{koordináta} \\ \text{érték} \end{array} \right)$

$\langle \text{koordináta irány} \rangle :: = X | Y | Z$
 $\langle \text{azonosítója} \rangle$

$\langle \text{koordináta} \rangle :: = \langle \text{érték} \rangle$

Az irányok azonosítására az ISO ajánlásnak megfelelően a konkrét szer-
számgeptől függően X, Y, Z stb. adatazonosítók használandók.

Az alappontok koordinátáit abszolút koordinátarendszerben kell megad-
ni, 0001 mm-es egységekben.

Az elmozdulások megadására szolgáló koordinátarendszert önmagával pár-
huzamosan akárhányszor el lehet tolni és az elmozdulásokat az ily módon
keletkezett rendszerben is meg lehet adni. /ld. 14-es rekord/.

A koordinátarendszer elforgatása jelenleg nem megengedett.

A vezérlési program minden újabb végrehajtásakor az első nullponttel-
tolásig egy alapkoordinátarendszer, nullponteltolástól újabb nullponttel-
tolásig az eltolt koordinátarendszer érvényes.

Az elmozdulások közül csak azokat kell megadni, amelyek irányában tény-
leges elmozdulás történik.

I, J, K adatok

I, J és K azonosítókkal körív interpolálása esetén a körív kezdőpontjá-
ból a körívhez huzott sugár hosszúságu érintőszakasz X, Y és Z irányu
vetületeit kell megadni.

Ha a vezérelt rendszer csak kétdimenziós interpolátort tartalmaz, és
így csak koordinátasíkban fekvő körívek interpolálására alkalmas, ak-
kor az I, J és K adatok közül mindig kettőt kell megadni, az aktuális
síknak megfelelően.

Előtolási sebesség megadása

F adatazonosító után a konstans pályamenti előtolási sebességet kell
megadni, mm/min, FR adatazonosító után mm/fordulat, FE adatazonosi-
tó után mm/fog egységekben.

F, FR, FE adatazonosítóval megadott előtolás értékek öröklődnek.

Gyorsmenet F, FR vagy FE adatazonosító után álló nulla értékkel programozandó. A gyorsmenet csak az adott mozgáselemre vonatkozik, az előző rekordra érvényes előtoló sebesség a következő, nem gyorsmenetű mozgáselemre automatikusan átöröklődik. Gyorsmenettel kör nem programozható.

A program első mozgáseleme előtt előtoló sebesség megadása kötelező.

Szerszámsugárkorrekció programozása

Mivel a MID-file a koordináta adatok között a kialakítandó kontur adatait kéri, ezért a feldolgozó program a MID file-ban megadott koordináta adatokból és a marószerszám sugarából illetve az esztergáló szerszám csucssugarának értékéből a szerszám referencia pont /középpont/ pályáját kiszámítja. Bizonyos esetekben ez a számítás igen munkaigényes, ezért a munka nehezét a MID-et előállító processzorra hárítjuk.

A MID-et feldolgozó program a szerszámsugár szerinti korrekciót a következőképpen végzi el:

A szerszám hosszának, a készülék méreteinek és a programozott nullponteltolásnak megfelelően korrigálja a pálya következő alappontjának programozott helyzetét.

Ha az aktuális pályaelem adatai között CRX, CRY vagy CRZ érték szerepel, akkor a vezérlés a szerszám adataiból kiolvasott R /sugár/ értékkel tovább korrigálja a következő alappont helyzetét. Körív típusu pályaelem esetén a körív középpontjának helyzetét megadó I, J, K adatokat is módosítja:

$$\begin{array}{ll} X' = CRX \cdot R + X ; & I' = CRI \cdot R + I_{MID} \\ Y' = CRY \cdot R + Y ; & J' = CRJ \cdot R + J_{MID} \\ Z' = CRZ \cdot R + Z ; & K' = CRK \cdot R + K_{MID} \end{array}$$

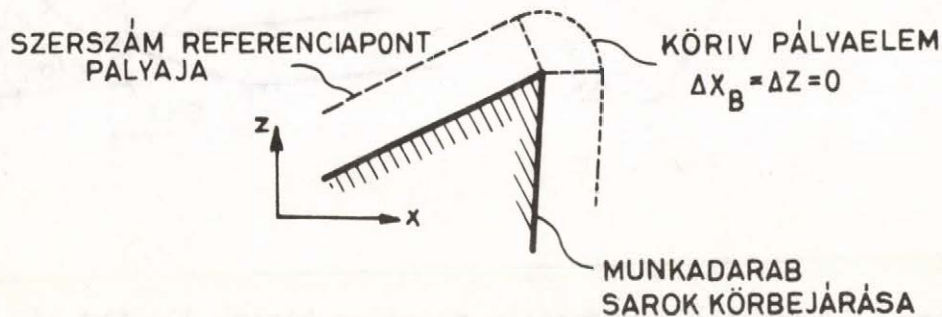
ahol X, Y, Z a megfelelő hosszkorrekciók beszámításával kapott, I_{MID} , J_{MID} , K_{MID} a programozott érték.

CRX, CRY, CRZ értékek mondatról mondatra nem öröklődnek. Nem megadott érték esetén a vezérlés nullát tételez fel.

Megjegyzés

Sarkok bejárásánál szükség lehet olyan pályaelemek /körívek/ programozására, amelyekben a ΔX , ΔY , ΔZ értékei mind nullák /a megmunkálási pont nem mozdul el, Fl.2. ábra/.

Az elmozdulást ilyenkor csak a CR értékek határozzák meg. A MID-ben ilyen pályaelemek megadása megengedett.



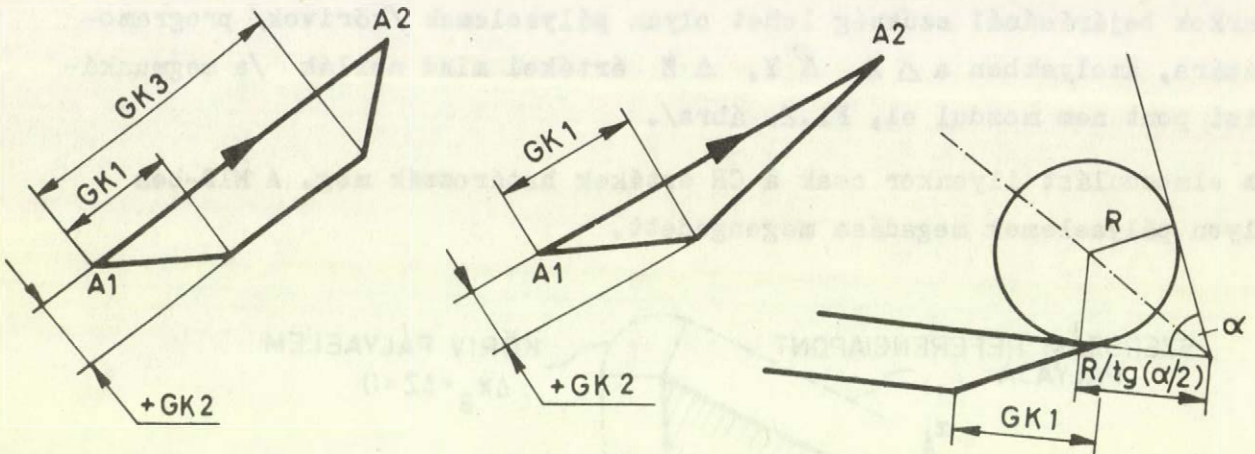
Fl.2 ábra: Sarkok körbejárása

Geometriai korrekció

A geometriai korrekciót csak a 2D megmunkálás esetére dolgoztuk ki. A korrekció mindig a megmunkálás síkjában értendő.

Hordósodás kompenzáció GK1, GK2, és GK3 adatazonosítókkal oldható meg /Fl.3 ábra/. A GK adatok a korrekciót a pályaelem kiindulási alappontjához képest adják meg. A hordósodás kompenzációt a vezérlés a szerszámsugárral korrigált pályán végzi el, ezért a ténylegesen megvalósuló korrekció a szándékolt korrekciótól eltér. Az eltérés kompenzálása céljából GK4 adatazonosítóval megadható $\pm 1 / \operatorname{tg}(\alpha / 2)$, ahol α a pálya két szomszédos eleme által bezárt szög. Az előjel /-/ hegyesszög, /+/ tompaszög esetén. A vezérlés $GK1' = GK + R \cdot GK4$ kifejezés szerint kompenzál, R a szerszám sugara.

Az alappontok korrekcióját KAX, KAY, KAZ adatazonosítók közül a megmunkálás síkjának megfelelő kettővel lehet előírni. Az alappontok korrekcióját a vezérlés a szerszámsugárkorrekció megkezdése előtt végzi el.



Fl.3. ábra: Hordosodás kompenzáció megadása három, illetve két paraméterrel

GK1 korrigálása a szerszámsugárral.

Előtolási sebesség korrekciója

FK adatazonosító után következő számérték az aktuális előtoló sebesség szorzótényezője. A vezérlés az aktuális sebességet korrigálja. Új előtoló sebesség beállítására a korábbi korrekció hatástalan.

Csavarmenet megmunkálása

Egybekezdésű menet: G33 kód esetén x, y, z a célkoordináták értékét adják /palást, sík és kupos menet lehetséges/, a menetemelkedés nagyságát a megfelelő I, J, K adatok tartalmazzák.

Pályabejárás pontossága befolyásának lehetőségei

FP adatazonosító egyirányú pozícióba érkezést ír elő. Negatív irányból való pozicionálás esetén a vezérlés automatikusan tulvezérel majd pozitív irányból pozícióba állít. Minden vezérelt tengelyre végrehajtódik. Nem öröklődő kód. Kör bejárására nem alkalmazható.

A kivárási feltételek között említett DS adatazonosító is a bejárasi pontosság befolyásolását szolgálja.

Programozás gépi koordinátarendszerben

XM, YM, ZM adatazonosítókkal megadott számértékek nem a megmunkálási pont, hanem a szán referenciapontjának helyzetét írják elő. Munkadarabcserénél, szerszámváltásnál használható. Kör menti bejárás nem programozható.

F1.2.3.4 Kapcsolási funkciók programozása

Kapcsolási utasításokat 02 típusu rekordokban lehet programozni.

A rekord felépítése:

$$\left[\begin{array}{cc} \langle \text{adatazonosító} \rangle & \langle \text{adat} \rangle \end{array} \right] \begin{array}{c} n \\ 0 \end{array}$$

$\langle \text{adat} \rangle :: = \langle \text{kód} \rangle | \langle \text{valós} \rangle$

$\langle \text{adatazonosító} \rangle :: = S | T | H | EL | TX | NM | SK | V | TC | VK$

F1.2.3.5 Kapcsolási rekordok adatazonosítóinak jelentése

Főorsó fordulatszám vezérlése: S adatazonosító után a fordulatszám értéke adandó meg. Kétirányú forgás lehetősége esetén a fordulatszám értéket a forgásirányra utaló jelzés követi /OJE, OJM/.

Főorsó leállítása: fordulatszám = 0.

Fokozatos főhajtás esetén a vezérlés a megadott értékhez legközelebb eső értéket kapcsolja.

V adatazonosítóval a vágósebesség értéke adható meg. A vezérlés a fordulatszámot az átmérő függvényében állítja be és szükség esetén /homlokesztergálásnál, pályamenti marásnál/ automatikusan korrigálja.

S, V értékek öröklődnek.

Főorső fordulatszám korrekció. SK adatazonosító után következő számérték az aktuális fordulatszám szorzótényezője. A vezérlés az aktuális fordulatszámot korrigálja. Új fordulatszám megadásakor a korábbi korrekció hatástalanná válik. A vágósebességgel megadott fordulatszám VK adatazonosító után álló adattal korrigálható.

Pozíció alapján azonosított szerszám beváltása. T adatazonosítót a revolverfej pozíció kódja követi. Az utasítás hatására a vezérlés szerszámcserét hajt végre. A pozíció szerinti azonosítással szemben a szerszám-típus szerinti azonosítást érdemes előnyben részesíteni.

Típus alapján azonosított szerszám kiválasztása, beváltása /04-es, 24-es típusu rekord/

Előváltás alatt a szerszámgép külső szerszámtartójának működtetését értjük, a beváltás egy szerszám megmunkálási helyzetbe hozása. A két művelet közül azt, amellyel a szerszámmal való tevékenység kezdődik, a konkrét szerszám kiválasztása előzi meg.

A kiválasztással együtt előváltást /TP/ vagy beváltást /TC/ is elő kell írni. A kiválasztást 04-es vagy 24-es rekordban kell programozni, attól függően, hogy a kapcsolódó előváltás illetve beváltás főfolyamatban vagy párhuzamos folyamatban zajlik. A már előváltott szerszám beváltása 02-es rekordban TC adatazonosítóval történik, mivel a beváltás kapcsolási funkció. A kiválasztáshoz szükséges adatok ilyenkor nem szerepelnek.

A lehetséges szekvenciákat és a megfelelő rekordtípusokat az Fl.4 ábra szemlélteti:

61-es rekord

Szubrutin definiálás vége

A szubrutin utolsó rekordja.

Paraméterekre hivatkozás a szubrutinok testében

A szerszámgépi művelő definiáló rekordokban a híváskor átadandó paraméterekre lehet hivatkozni.

A hivatkozás a PV adatazonosítóval és egy sorszámmal történik. A sorszám mondja meg, hogy a híváskor átadott paraméter lista hányadik elemére kívánunk hivatkozni.

Paraméterek deklarációja

A szubrutin paramétereinek típusát a rutin második rekordjában /vagyis a 60-as rekord után/ deklarálni kell. A rekord típusa 50, felépítését a fej rekordjai között már tárgyaltuk.

Relatív koordinátarendszer használata

Szubrutinok testében relatív koordinátarendszert is lehet használni. A relatív koordinátarendszer középpontja a pálya előző alappontja, tengelyei párhuzamosak az alapkoordinátarendszer tengelyeivel. A relatív koordinátákat DX, DY, DZ adatazonosítók jelölik.

F1.2.4.2 Szubrutinok hívása

Szubrutin hívása 36-os típusu rekorddal történik. A rekord felépítése:

$$\langle \text{szubrutin} \rangle \quad \left[\begin{array}{c} \langle \text{paraméter} \rangle \\ \text{azonosító} \end{array} \right]_0^n$$

$\langle \text{paraméter} \rangle :: = \langle \text{konstans} \rangle \mid \langle \text{MID változó} \rangle$

A hívott szubrutin végrehajtását kezdeményezi.

A rutin végrehajtása után a megmunkálás a következő rekorddal folytatódik.

Fl.2.4.3 Műveletelem típusu szubrutinok

A 4. fejezetben említettük, hogy a vezérlési nyelv műveletelem típusu szubrutinjainak írására speciális nyelvet definiáltunk. A nyelv blokk-strukturájú, az ALGOL szintaksziszra építettünk.

A műveletelem típusu szubrutinok írására definiált nyelvbe a hagyományos vezérlő nyelvek elemei /ALGOL/ standard függvényként kerültek beépítésre. Az új standard függvények azonosítói megegyeznek a MID nyelv adatazonosítóival. A függvények működésének lényege, hogy a paraméterként kapott értéket a megfelelő adatazonosítóval megfejeelve az aktuális output mondatba helyezik. Különleges függvényazonosító a H, amely hatására az aktuális mondat terminálódik és az implementációtól, üzemmódtól függően vagy végrehajtódik vagy output hordozóra kerül.

A műveletelem típusu szubrutinok programozási nyelve

A MID változók, a paraméteres mozgásmegadás és a MID változók /Fl.2.3.8-ban ismertetett/ értékadási lehetőségei már majdnem elegendők a műveletelem típusu szubrutinok írásához. Két bővítést kell csupán tennünk: meg kell engedni változók tömbjének használatát és be kell vezetni a szubrutin lokális paramétereit. A programozási munka megkönnyítésére bevezetjük még az IF ...THEN...ELSE konstrukciót, a ciklus utasítást és az összetett utasítást. A nyelv részletes szintaksziszát függelékben közöljük.

Műveletelem típusu szubrutinok integrálása a számítógépes szerszámgép-irányítási rendszerbe

Új műveletelem típusu szubrutin felgenerálása a rendszerbe abból áll, hogy egy fordítóprogrammal a forrásnyelvi szöveget a run-time környezet által érthető formátumba konvertálják, majd könyvtárazzák. A számítógépes szerszámgépvezérlés a műveletelem típusu szubrutinokat ebben a bináris, tehát tömör formában tárolja. Maga a fordítóprogram nem része a vezérlésnek és esetleg nem is a vezérlő számítógépen fut /egyedi vezérlés esetében semmi esetre sem/.

A bináris formára fordított szubrutin fizikailag akár egy olcsó, állandó memóriában is tárolható.

Műveletelem típusu szubrutin hívásakor aktivizálódik az említett run-time környezet, átveszi a paramétereket és végrehajtja a programot. A szubrutin munkájának befejeződése után a vezérlés a hagyományos run-time programnak adódik vissza az alkatrészprogram értelmezésének folytatása céljából. ALGOL terminológiában fogalmazva, a műveletelem típusu szubrutin hívó utasítása automatikusan olyan programmá egészül ki, amelynek egyetlen utasítása maga a hívó utasítás. Ez a program hajtódik végre, majd visszatérés történik az eredeti nyelv interpretálására.

Implementációs kérdés, de érinti az egész koncepció gazdaságosságát, és ezért megemlítjük, hogy a két run-time környezet nem diszjunkt. A műveletelem-típusu szubrutinok run-time környezete a megszokott run-time környezet bővítése. A nyelv reprezentációja felől megközelítve ez azt jelenti, hogy mindkét nyelv belső reprezentációja közös.

F2. Függelék: Technológiai típusu rekordok formális leírása

A függelékben a MID nyelv mozgási és kapcsolási rekordjainak /02 és 03 típus/ formális leírását közöljük. A 02 és 03 rekordokban használt adatazonosítók közül azoknak az értelmezését közöljük, amelyekre a Csepeli Szerszámgépgyár ERI-250 tárcsa esztergájának programozására szükség van.

F2.1 A módszer

A leírás módszere [161] továbbfejlesztett változata. Lényege a következő. Transzformációs szabályokat közlünk, amelyek megadják, hogy a szabály baloldalán leírt állapot fennállása esetén milyen új állapot áll be. Az új állapotot a szabály jobboldala tartalmazza. A jobb és baloldalt "→" választja el. Az alkatrészprogram végrehajtása /a megmunkálás/ a baloldalával érvényesített szabály alkalmazásából áll.

Csak főfolyamatból álló megmunkálás esetén minden időpillanatban legfeljebb egy szabály alkalmazásának feltétele teljesül. Párhuzamos folyamat lehetősége esetén ez a szabály triviálisan módosul. Ha az adott pillanatban egy szabály sem alkalmazható, akkor a vezérlés a végrehajtással kivár. A kivárás időtartama alatt zajlanak a szerszámgépen a tulajdonképpeni megmunkálási folyamatok. Ezek befejeződése a szerszámgép állapotának megváltozásával jár, aminek következtében valamelyik transzformációs szabály alkalmazhatóvá válik.

A pillanatnyi állapotnak két komponense van, nevezetesen a program értelmezésének, valamint a szerszámgépnek és a vezérlésnek az állapota.

A program értelmezés állapota a vezérlő program azon pozíciója, amelyenél a program értelmezés éppen tart. A program alapelemeinek sorozata. A programértelmezés állapotát program alapelemek közé szurt speciális jellel, markerrel írjuk le.

A marker előtt álló alapelem-sorozat a következő állapotváltozást nem befolyásolja, ezért jelenlétét nem jelezzük.

A szerszámgép és a vezérlés állapota vektor. A vektor elemeinek értékészlete az elemtől függ /valós szám, egész kód, logikai érték, vektor stb./, ezért elemenként definiáljuk. A vektorelemekre szimbólikus kód-

dal lehet hivatkozni. A szerszámgép és a vezérlés állapotát az összes vektorelem helyett csak azon vektorelemekkel adjuk meg, amelyeknek értéke az adott pillanatban nem közömbös, illetve amelyik megváltozik.

A transzformációs szabályok baloldalán tehát a program olvasás állapota és az adott szabály végrehajthatóságához lényeges szerszámgép és vezérlés állapotok állnak azon értékeik megjelölésével, amelyek fennállása esetén a szabály végrehajtódik. A baloldal összetevői logikai függvényekbe komponálhatók.

A transzformációs szabály jobboldala az automata új állapotát írja le.

A jobboldal ugyanolyan összetevőkből áll, mint a baloldal, csak itt a konkrét értékek azt jelölik ki, amelyet az adott vektorelem a szabály alkalmazásakor felvesz. A jobboldal elemeit pontosvessző választja el egymástól, amely logikai "és" kapcsolatot jelöl.

F2.2 Jelölések

◇: a vezérlő program értelmezése az adott helynél tart;

§, §: a vezérlő program alapelemeinek tetszőleges sorozata ;

⟨REAL⟩, ⟨INT⟩, ⟨CODE⟩, ⟨ID⟩: tetszőleges valós szám, egész, kód, illetve azonosító, amelyet a vezérlés az alkatrészprogramban talál és amelynek értékére a transzformációs szabály jobboldalán történik hivatkozás ;

∧, ∨, ¬: logikai és, vagy, nem viszony;

CLS = N: a szerszámgép "hűtés" jellemzőjének aktuális értéke N ;

SF := /REAL 1 * SS/: a vezérlés SF jellemzőjének értéke /amely egy valós szám, ld. F2.3/ az alkatrészprogramban talált REAL 1 valós szám és a szerszámgép SS jellemzője aktuális értékének szorzata.

F2.3 Szerszámgép és vezérlés állapotvektor komponensei

Az állapotvektor összetevőinek értékkészletét definiáljuk. A jobboldali oszlopban minden összetevőre megadjuk azt a szimbólikus kódot, amellyel az adott összetevőre a transzformációs szabályokban hivatkozni lehet.

/1/	< hűtés >	:: =	Y N	CLS
/2/	< főorsó >	:: =	N < valós >	SS
/3/	< előtoló mozgás >	:: =	Y N	MTS
/4/	< foglaltság >	:: =	Y N	BYS
/5/	< beváltott szerszám >	:: =	< szerszám >	TS
	< szerszám >	:: =	< típus > < egyedi kód > < élek száma > < korrekciók > < éltartamok >;	

ahol: < típus > :: = < azonosító >

< egyedi kód > :: = < kód >

< élek száma > :: = < egész >

< éltartamok > :: = $\left[\begin{array}{c} \text{élek száma} \\ \text{élek száma} \end{array} \right]$ < egész >

< korrekciók > :: = $\left[\begin{array}{c} \text{élek száma} \\ \text{élek száma} \end{array} \right] \begin{array}{c} \text{hosszkorrekció} \\ \text{sugár-} \\ \text{korrekció} \end{array}$

< hosszkorrekció > :: = < valós > < valós > LX, LZ

< sugárkorrekció > :: = < valós > R

/6/	< pályaelem >	:: =	N LIN CCW CW TAP	IPS
/7/	< elmozdulás >	:: =	Y N	MVS
/8/	< gyorsmenet >	:: =	Y N	RDS
/9/	< előtolás >	:: =	N < valós >	SF

/10/	< megmunkálási végpont koordinátája >	:: = N < valós >	STX, STZ
/11/	< pálya segédadata >	:: = N < valós >	SI, SK
/12/	< geometriai korrekció >	:: = < valós >	SGK1, SGK2, SGK3
/13/	< sugárkorrekció >	:: = < valós >	SCRX, SCRZ, SCRI, SCRK
/14/	< rekord típus >	:: = N < kód >	TRS
/15/	< vezérelt pont pillanatnyi koordinátája >	:: = N < valós >	SRFX, SRFZ
/16/	< vezérelt végpont >	:: = N < valós >	SRTX, SRTZ
/17/	< megmunkálási pont pillanatnyi koordinátája >	:: = < valós >	SFX, SFZ
/18/	< változó állapota >	:: = VAR (< azonosító >)	

Megjegyzés:

A "foglaltság" komponens "N" értékű, ha a vezérlés az előzőekben kiadott parancsok korrekt végrehajtását jelezte. /A "foglaltság" ellentettje a "készenlét"/.

Megjegyzés:

A beváltott szerszám X irányu hosszkorrekciós értékére TS /LX/ jelöléssel lehet hivatkozni.

Megjegyzés:

/3-hoz/ : v.ö: Fl.2.1 pont, DS adatazonosító

Megjegyzés:

/18-hoz/: A MID változók pillanatnyi állapotára VAR (<AZONOSÍTÓ>) jelöléssel lehet hivatkozni.

Megjegyzés:

/10, 15, 16, 17-hez/: A megmunkálási végpont a megmunkálási pontnak a rekord végrehajtása utáni helyzete /az X, Z adatazonosítókkal progra-

mozott érték/. XM, ZM adatazonosítókkal a szán referenciapontja következő helyzetét adjuk meg. Minden mozgási rekord megkezdésekor az előző rekordban beprogramozott értékű a vezérelt pont és a megmunkálási pont koordinátája, feltéve, hogy az előző rekord végrehajtása normálisan befejeződött.

F2.4 Rekordstruktúra

$$\langle \text{MID rekord} \rangle :: = \langle \text{tipus} \rangle \langle \text{tipus módosító} \rangle \left[\langle \text{kivárási feltétel} \rangle \right]^1_0$$

$$\langle \text{adatok} \rangle \langle \text{rekordvége} \rangle$$

$\langle \text{tipus} \rangle :: = \langle \text{kód} \rangle$

$\langle \text{tipus módosító} \rangle :: = \text{FN} \mid \text{FT} \mid \text{E}$

$\langle \text{kivárási feltétel} \rangle :: = \text{DV} \langle \text{azonosító} \rangle \mid \text{DT} \langle \text{real} \rangle \mid \text{DP} \mid \text{DS}$

$\langle \text{adatok} \rangle :: = \text{rekord tipustól függ, ld. 3. fejezet}$

$\langle \text{rekord vége} \rangle :: = \text{implementációtól függő jelzés}$

/1/ $\langle \text{rekord vége} \rangle \Diamond \langle \text{CODE} \rangle \text{FN} \xi \rightarrow \langle \text{CODE} \rangle \text{FN} \Diamond \xi ; \quad \text{TRS} := \langle \text{CODE} \rangle$

/2/ $\langle \text{rekord vége} \rangle \Diamond \langle \text{CODE} \rangle \text{FT} \xi \wedge \text{BYS} = \text{N} \rightarrow \langle \text{CODE} \rangle \text{FT} \Diamond \xi ; \quad \text{TRS} := \langle \text{CODE} \rangle$

/3/ $\langle \text{rekord vége} \rangle \Diamond \langle \text{CODE} \rangle \text{E} \text{DV} \langle \text{ID} \rangle \xi \wedge \text{VAR}(\langle \text{ID} \rangle) = \text{Y} \rightarrow \langle \text{CODE} \rangle \text{E} \text{DV} \langle \text{ID} \rangle \Diamond \xi ;$
 $\text{TRS} := \langle \text{CODE} \rangle$

/4/ $\langle \text{rekord vége} \rangle \Diamond \langle \text{CODE} \rangle \text{E} \text{DP} \xi \wedge \text{VAR}(\text{R2S}) = \text{Y} \wedge \text{BYS} = \text{N} \rightarrow \langle \text{CODE} \rangle \text{E} \text{DP} \Diamond \xi ;$
 $\text{TRS} := \langle \text{CODE} \rangle$

/5/ $\langle \text{rekord vége} \rangle \Diamond \langle \text{CODE} \rangle \text{E} \text{DS} \xi \wedge \text{MTS} = \text{N} \rightarrow \langle \text{CODE} \rangle \text{E} \text{DS} \Diamond \xi ;$
 $\text{TRS} := \langle \text{CODE} \rangle$

Megjegyzés:

R2S a párhuzamos folyamat foglaltsági jele. ERI-250 szerszámgépen nem fordul elő, megfelelő transzformációs szabályt csak a teljesség kedvéért közöltük.

F2.5 Egyéb transzformációs szabályok

Pályaelem

/6/ $\diamond X \langle \text{REAL} \rangle \rightarrow X \text{ REAL} \diamond$; STX := $\langle \text{REAL} \rangle$; MVS: = Y
 /7/ $\diamond Z \langle \text{REAL} \rangle \rightarrow Z \text{ REAL} \diamond$; STZ := $\langle \text{REAL} \rangle$; MVS: = Y
 /8/ $\diamond I \langle \text{REAL} \rangle \rightarrow I \text{ REAL} \diamond$; SI := $\langle \text{REAL} \rangle$; MVS: = Y
 /9/ $\diamond K \langle \text{REAL} \rangle \rightarrow K \text{ REAL} \diamond$; SK := $\langle \text{REAL} \rangle$; MVS: = Y
 /10/ $\diamond XM \langle \text{REAL} \rangle \rightarrow XM \text{ REAL} \diamond$; SRT X := $\langle \text{REAL} \rangle$; MVS: = Y
 /11/ $\diamond ZM \langle \text{REAL} \rangle \rightarrow ZM \text{ REAL} \diamond$; SRTZ := $\langle \text{REAL} \rangle$; MVS: = Y
 /12/ $\diamond GO1 \rightarrow GO1 \diamond$; IPS := LIN
 /13/ $\diamond GO2 \rightarrow GO2 \diamond$; IPS := CW
 /14/ $\diamond GO3 \rightarrow GO3 \diamond$; IPS := CCW
 /15/ $\diamond G33 \rightarrow G33 \diamond$; IPS := TAP
 /16/ $\diamond \langle \text{KORR} \rangle \langle \text{REAL} \rangle \rightarrow \langle \text{KORR} \rangle \langle \text{REAL} \rangle \diamond$; $\langle \text{KORRS} \rangle := \langle \text{REAL} \rangle$
 $\langle \text{KORR} \rangle :: = GK1 \mid GK2 \mid GK3 \mid CRX \mid CRZ \mid CRI \mid CRK$
 $\langle \text{KORR S} \rangle :: = SGKL \mid SGK2 \mid SGK3 \mid SCRX \mid SCRZ \mid SCRI \mid SCRK$

Előtolási sebesség

/17/ $\diamond FO \rightarrow FO \diamond$; RDS := Y
 /18/ $\diamond FR O \rightarrow FR O \diamond$; RDS := Y
 /19/ $\diamond F \langle \text{REAL} \rangle \rightarrow F \langle \text{REAL} \rangle \diamond$; SF := $\langle \text{REAL} \rangle$
 /20/ $\diamond FR \langle \text{REAL} \rangle \rightarrow FR \langle \text{REAL} \rangle \diamond$; SS := $(\langle \text{REAL} \rangle * SS)$

Főrsó

$$\begin{aligned} /21/ \quad \Diamond S \langle \text{REAL} \rangle \nrightarrow S \langle \text{REAL} \rangle \Diamond \nrightarrow; \quad SS &:= \langle \text{REAL} \rangle \\ /22/ \quad \Diamond V \langle \text{REAL} \rangle \nrightarrow V \langle \text{REAL} \rangle \Diamond \nrightarrow; \quad SS &:= (\langle \text{REAL} \rangle / (2 \pi \cdot \text{SFX})) \end{aligned}$$

Hütés

/23/ \diamond M08 \rightarrow M08 \diamond \rightarrow ; CLS := Y
/24/ \diamond M09 \rightarrow M09 \diamond \rightarrow ; CLS := N

Szer szám

/25/ ◇ TC < speci-
fikáció > < rekord vége > → TC < specifi-
káció > < rekord vége > ◇ ;

TS : = f (specifikáció)

Rekord vége

/26/ $\Diamond \langle \text{rekord vége} \rangle \} \wedge \text{TRS} = 03 \wedge \text{MVS} = Y \wedge \text{IPS} = (7N) \wedge$
 $(\text{SF} = \langle \text{REAL} \rangle \vee \text{RDS} = Y) \rightarrow$
if $\text{STX} = (7N)$ then $\text{SRTX} := (\text{STX} + \text{TS}(\text{LX}) + \text{SCRX} * \text{TS}(\text{R}));$
if $\text{STZ} = (7N)$ then $\text{SRTZ} := (\text{STZ} + \text{TS}(\text{LZ}) + \text{SCRZ} * \text{TS}(\text{R}));$

MTS : = Y ;
BYS : = Y ;
SCRX : = 0 ; SCRZ : = 0 ; SCRI : = 0 ; SCRK : = 0 ;
SGK1 : = 0 ; SGK2 : = 0 ; SGK3 : = 0 ;
MVS : = N ;
RDS : = N ;
TRS : = N

/27/ \Diamond <rekord vége> \wedge TRS = 0 2 \rightarrow <rekord vége> \Diamond ; BYS : = Y ;
TRS : = N

F2.6 Szöveges megjegyzések a transzformációs szabályokhoz

A rekordstruktúrára vonatkozó szabályok /1, 2, 3, 4, 5/ azokat a feltételeket adják meg, amelyeket teljesülése esetén egy adott rekord végrehajtása megkezdődhet. A rekord típusát a vezérlés TRS jellemzője tárolja, erre a jellemzőre a /26, 27/ szabályokban történik hivatkozás.

A 6-os és 7-es szabályok azt írják le, hogy az X adatazonosítóval programozott érték a megmunkálási pont rekord utáni helyzete. MVS azt jelzi, hogy az adott rekordban programoztak-e elmozdulást.

A 20-as szabály azt a kifejezést tartalmazza, amellyel a fordulatonkénti előtolásból a tényleges előtolás meghatározható. Hasonlóképpen a 22-es szabály a vágósebességből a fordulatszám értékének meghatározását mutatja.

Az ERI-250 gépen a szerszám kiválasztás és beváltás egyszerre történik, TC adatazonosítóval megadott típus alapján.

A mozgási rekord végén /26-os szabály/ történik meg a vezérelt pont végrehajtás utáni koordinátájának kiszámítása és a vezérlés megfelelő állapotainak beállítása, a mozgás megkezdése. A mozgás végeztével a megmunkálási pont és a vezérelt pont pillanatnyi koordinátái /SFX, SFZ, SRFX, SRFZ/ a megfelelő végpontok koordináta értékeit veszik fel /STX,

STZ, SRTX, SRTZ/. Ezután $MTS = N$, $BYS = N$ érték áll be, ami új transzformációs szabály alkalmazását engedélyezi.

A kapcsolási rekordok végén /27-es szabály/ a vezérlés BYS állapota Y értéket vesz fel. N állapot a megfelelő funkciók végrehajtása után áll be.

F3. Függelék: Műveletelem típusu technológiai alprogramok programozási nyelve

A függelék az Fl.2.4.3 szerinti gazda-nyelv szintakszisát tartalmazza, részben közvetlenül a kibővített metanyelven, részben [162] által bevezetett táblázatos formában. A kifejezések szintakszisát szövegszerűen írjuk le. A leírást magyarázó megjegyzések egészítik ki.

/1/ $\langle \text{program} \rangle ::= \langle \text{egyszerű blokk} \rangle$

/2/ $\langle \text{egyszerű blokk} \rangle ::= \begin{array}{l} \underline{\text{begin}} \left[\langle \text{deklaráció} \rangle \right]_0^n \\ \langle \text{eljárás utasítás} \rangle \quad \underline{\text{end}} \end{array}$

/3/ $\langle \text{eljárás utasítás} \rangle ::= \langle \text{egyszerű eljárás} \rangle | \begin{array}{l} \langle \text{minta-típusu} \\ \text{eljárás uta-} \\ \text{sitás} \end{array}$

/4/ $\langle \text{egyszerű eljárás} \rangle ::= \langle \text{azonosító} \rangle \left[\langle \text{aktuális paraméterek} \rangle \right]_0^n$

/5/ $\langle \text{aktuális paraméterek} \rangle ::= \langle \text{ar.kif.} \rangle \left[\langle \text{ar.kif} \rangle \right]_0^n | \langle \text{üres} \rangle$

/6/ $\begin{array}{l} \langle \text{minta-típusu} \\ \text{eljárás utasítás} \end{array} ::= \text{PATTERN} \left(\begin{array}{l} \langle \text{egyszerű eljárás} \rangle, \\ \langle \text{eljárás utasítás} \rangle \end{array} \right)$

Megjegyzés / 5-höz/: ar.kif rövidítés; aritmetikai kifejezés helyett áll.

Megjegyzés /2-höz/: Az egyszerű blokk csak az eljárás utasításban használt aktuális paraméterek deklarációját tartalmazza, az eljárás deklarációját a rendszer a könyvtárából veszi.

Valójában a program írása nem a gazda-nyelven történik, hanem a /MID nyelvű/ programtörzs azon utasításai, amelyek műveletelem típusu alprogramokat hívnak, gazda-nyelvi programokká alakulnak és végrehajthatódnak. Az aktuális paraméterek a MID nyelvű programban deklaráltak.

Megjegyzés /4-hez/: Csak érték szerinti átadás megengedett. Eredmény visszadás változó típusu aktuális paramétereken keresztül lehetséges /ennek a lehetőségnek a kihasználása nem lesz gyakori/.

Megjegyzés /6-hoz/: Szintaktikailag valóban egyszerű eljárás áll, valójában csak un. pattern-típusu eljárások hívhatók. A pattern típusu eljárások különlegessége: tartalmazznak CALL utasítást, nem hívhatnak eljárást.

/7/ $\langle \text{aritmetikai kifejezés} \rangle :: =$ lehet egész vagy valós típusu.

Felépítését tekintve lista, amelyet a következő elemek komponálnak:

- számok
- változók
- műveleti jelek (+ - * /)
- kerek zárójelek
- standard függvények
/sin, cos, tan, abs, sqrt, arctan, ln, exp, entier/

A számok és változók típusa meg kell egyezzen a kifejezés típusával.

Megjegyzés: a standard függvények argumentuma aritmetikai kifejezés, amely nem tartalmazhat standard függvényeket.

/8/ $\langle \text{logikai kifejezés} \rangle :: = \langle \text{aritm.kif.} \rangle \langle \text{reláció} \rangle \langle \begin{smallmatrix} \text{aritm.} \\ \text{kif.} \end{smallmatrix} \rangle$

/9/ $\langle \text{reláció} \rangle :: = \langle | \leq | = | \neq | > | \rangle$

/10/ $\langle \text{kifejezés} \rangle :: = \langle \text{aritmetikai kifejezés} \rangle | \langle \text{logikai kifejezés} \rangle$

/11/ $\langle \text{deklaráció} \rangle :: = \langle \text{egyszerű változó dekl.} \rangle | \langle \text{tömb dekl.} \rangle |$
 $\langle \text{eljárás dekl.} \rangle$

/12/ $\langle \text{egyszerű változó dekl.} \rangle :: = \langle \text{típus} \rangle \langle \text{azonosító} \rangle \left[\langle \text{azonosító} \rangle \right]^n$

/13/ $\langle \text{tömb deklarációja} \rangle ::= \langle \text{típus} \rangle \text{array} \langle \text{azonosító} \rangle [\langle \text{egész} \rangle]$

/14/ $\langle \text{típus} \rangle ::= \text{integer} \mid \text{real} \mid \text{boolean}$

/15/ $\langle \text{eljárás deklarációja} \rangle ::= \text{procedure} \langle \text{azonosító} \rangle (\langle \text{paraméterek} \rangle);$

$$\left[\langle \text{specifikáció} \rangle \right]_0^n \langle \text{törzs} \rangle$$

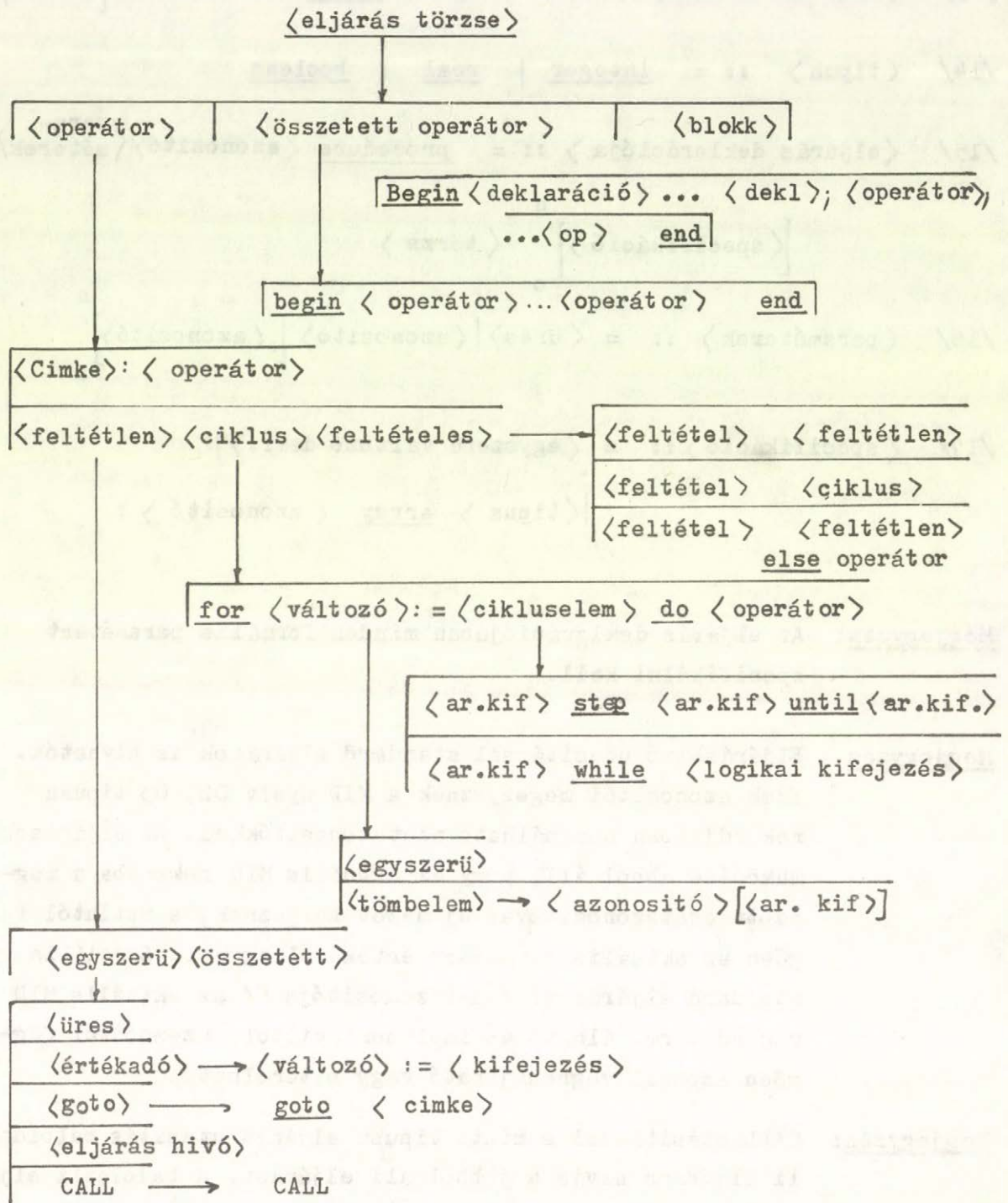
/16/ $\langle \text{paraméterek} \rangle ::= \langle \text{üres} \rangle \mid \langle \text{azonosító} \rangle [\langle \text{azonosító} \rangle]_0^n$

/17/ $\langle \text{specifikáció} \rangle ::= \langle \text{egyszerű változó dekl.} \rangle \mid$
 $\mid \langle \text{típus} \rangle \text{array} \langle \text{azonosító} \rangle ;$

Megjegyzés: Az eljárás deklarációjában minden formális paramétert specifikálni kell.

Megjegyzés Eljáráshívó utasítással standard eljárások is hívhatók. Ezek azonosítói megegyeznek a MID nyelv 02, 03 típusu rekordjaiban használható adatazonosítókkal. Az eljárások működése abból áll, hogy az aktuális MID rekordba a megadott adatazonosítóval új mezőt helyeznek, a rutintól függően az aktuális paraméter értékével együtt. Speciális standard eljárással /adatazonosítója H/ az aktuális MID rekord terminálható és implementációtól, üzemmódtól függően azonnal végrehajtható vagy eltárolható.

Megjegyzés: CALL utasítással a minta típusu eljárás utasítás baloldali eljárása hívja a jobboldali eljárást. A baloldali eljárás rendszerint ciklus, amelynek ciklusmagjában található a CALL utasítás.

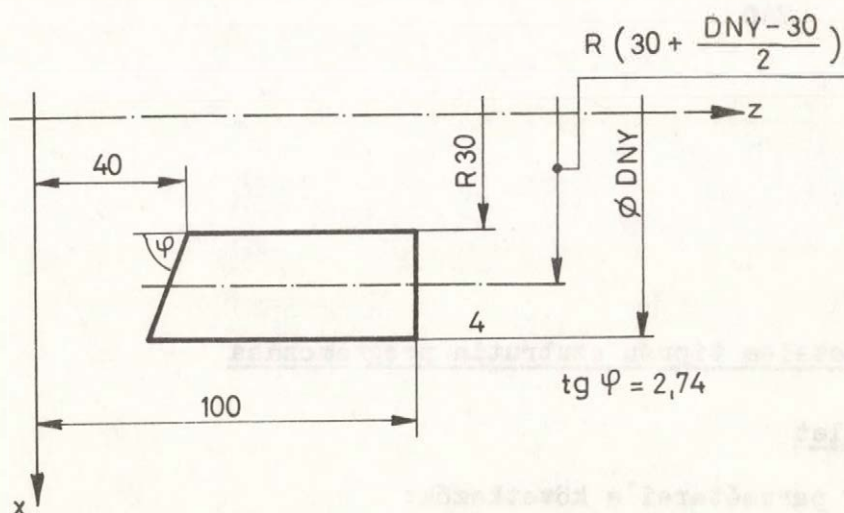


.F3.1 ábra: Fogadónyelv szintakszis táblája

F4. Függelék: Programozási példák

F4.1 Paraméteres mozgásprogramozás MID nyelvű alkatrészprogramban

Esztergálási példát mutatunk be, amely a paraméteres mozgásprogramozás lehetőségét illusztrálja. Az F4.1 ábra az eltávolítandó területet mutatja. E4 helyzetét X irányban a nyersdarab tényleges átmérője határozza meg. A nyersdarab szórása akkora, hogy a terület eltávolítása két fogással mindenképpen megoldható. Olyan programot mutatunk be, amelynek végrehajtása során az első fogás a tényleges eltávolítandó terület közepére áll be.



F4.1 ábra: Esztergálási feladat

A programot a MID nyelv egy olyan reprezentációjában fogalmazzuk meg, amely kézi programozásra alkalmas. A reprezentációt nem definiáljuk, de a MID nyelv fenti ismertetése alapján a példa program megértése nem okozhat nehézséget.

REAL

DNY

.

.

.

KEZELŐI - DIALÓGUS "NYERSDB ATM ? " , DNY

.

.

.

GYORS Z102 $\times (15 + \text{DNY} / 4)$

V400 $Z (40 - (\text{DNY}/4-15) / 0.35)$

GYORS DX2

GYORS Z102

GYORS X30

V400 Z40

.

.

.

F4.2 Műveletelem típusu szubrutin programozása

Furási művelet

A szubrutin paraméterei a következők:

L : A furat mélysége

D : A szerszám átmérője

FEEDS : A FEED tömb hossza

FEED : Valós tömb, az előtolási sebességek értékeit tartalmazza mm/ford egységekben. Az első előtolási szakaszt a tömb első elemének, a másodikat a másodiknak, stb., megfelelő előtolási sebességgel teszi meg. Ha a tömbelemek elfogynak, az összes többi szakaszt az utolsó elemnek megfelelő sebességgel hajtja végre

BREAK : Technológiai szempontból szükséges kiemelés mértéke

RP : A furó kezdeti távolsága a megmunkálandó felületről. Visszatérés ugyanebbe a pozícióba történik.

A rutin[164]-ben leírt módon meghatározza az adott szerszámméretnek megfelelő előtolási szakasz hosszakat. Minden szakasz után BREAK-nek megfelelő értékkel kiemeli a szerszámot.

Hasonló jellegű furási műveletek könnyen konstruálhatók, [163].

A rutin belső változóinak jelentése:

NUMS : Az adott furat megmunkálásához szükséges kiemelések száma, valós ;

NUM : A kiemelések száma, NUMS-ból felfelé kerekítéssel kapott egész ;

LI : Az aktuális előtolási szakasz hossza ;

POZ : A szerszám pillanatnyi pozíciója a furat felszínéhez képest.

```
SUBROUTINE DRILL1(L, D, FEEDS, FEED, BREAK, RP);
  REAL L, D, BREAK, RP;
  INTEGER FEEDS;
  REAL ARRAY FEED;
  BEGIN INTEGER I, K, NUM;
    REAL LI, POZ, NUMS;
    POZ:=0;
    IF D<=15.0 THEN NUMS:=(0.0022*D+0.087)*L+(0.4-0.48*D)
      ELSE NUMS:=(0.1095-0.00098*D)*L-
        (0.029*D-5.266);
    NUM:=ENTIER(NUMS+1);
    IF NUM>1 THEN
      BEGIN FOR I:=1 STEP 1 UNTIL NUM-1 DO
        BEGIN IF D<=15.0 THEN LI:=(1-0.4+0.48*D)/
          (0.0022*D+0.087)
          ELSE LI:=(1+5.266+0.029*D)/
          (0.1095-0.00098*D);
          IF I=1 THEN DZ(LI+RP) ELSE DZ(LI+BREAK);
          POZ:=POZ+LI;
          IF I=>FEEDS THEN V(FEED[FEEDS])
            ELSE V(FEED[I]); H;
          RAPID; DZ(-BREAK); H;
        END;
        DZ(L-POZ);
        IF I=>FEEDS THEN V(FEED[FEEDS]) ELSE V(FEED[I]);
        H;
      END ELSE
        BEGIN DZ(L+RP); V(FEED[I]); H;
        END;
        RAPID; DZ(-(L+RP)); H;
  END
```

END

* F4.2. ábra: furóciklus programja

F4.3 Műveletelem típusu szubrutin programozása:

furóciklus és alkalmazása

A furóciklus pattern-típusu eljárás. Paramétereinek jelentése a következő:

- DIRX, DIRY : két valós szám, amely egy XY síkbeli irányt jelöl ki ;
NINCRS : INCR tömbben tárolt számpárok száma ;
INCR : valós tömb, amely lépéstávolságokat tartalmaz. Minden lépéstávolságot számpár ír elő: a számpár első eleme ismétlési tényező /értéke > 1 , második eleme maga a lépéstávolság.

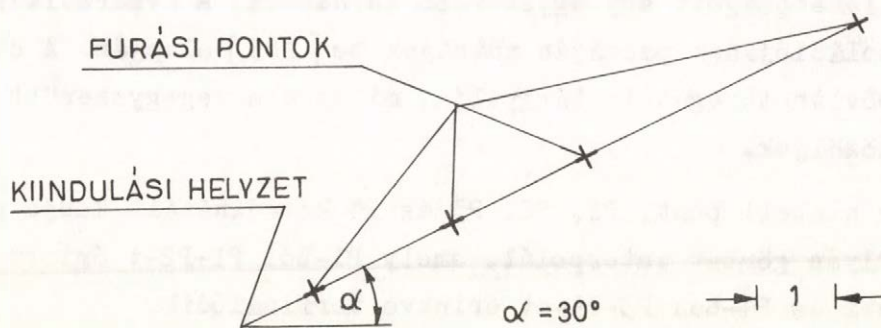
Az eljárás működése abból áll, hogy a pillanatnyi helyzetből kiindulva a megadott irányban az aktuális lépéstávolságnak megfelelő hosszban gyorsmenettel elmozdit, majd meghívja a minta-típusu eljárás utasítás másik elemét. A működés az INCR tömb kimerülésekor fejeződik be.

```
SUBROUTINE PTRN(DIRX, DIRY, NINCRS, INCR);  
  REAL DIRX, DIRY;  
  REAL ARRAY INCR;  
  INTEGER NINCRS;  
  BEGIN REAL C1, C2, S1, S2, DTX, DTY;  
    INTEGER I, J, K;  
    S1:=ABS(DIRX); S2:=ABS(DIRY); DTX:=0; DTY:=0;  
    DIRX:=S1/DIRX; DIRY:=S2/DIRY;  
    C1:=DIRX/SQRT(1+S2*S2/(S1*S1));  
    C2:=DIRY*C1*S2/S1;  
    FOR I:=1 STEP 1 UNTIL NINCRS DO  
      BEGIN J:=2*I-1;  
        S1:=C1*INCR[J+1];  
        S2:=C2*INCR[J+1];  
        FOR K:=1 STEP 1 UNTIL INCR[J] DO  
          BEGIN DTX:=DTX+S1; DTY:=DTY+S2;  
            DX(S1); DY(S2); F(0); H;  
            CALL  
          END  
        END  
      END  
    F(0); DX(-DTX); DY(-DTY); H  
  END  
END
```

*

Hivási példák:

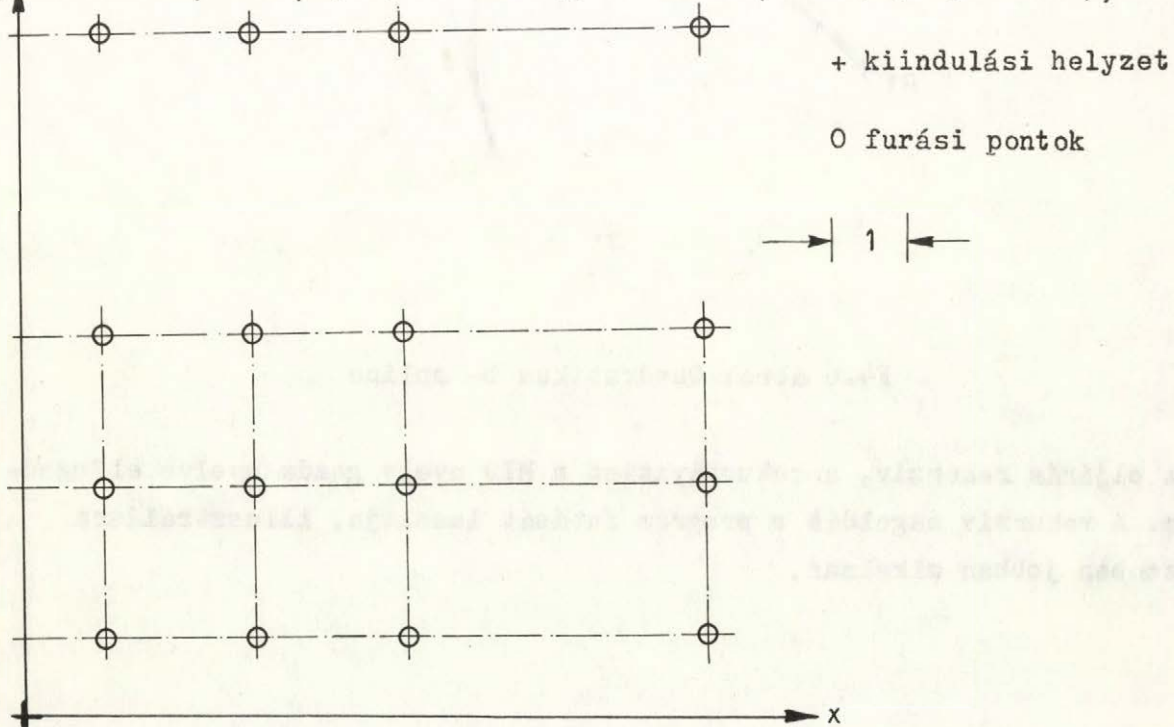
1/ INCR : 1,2,2,4
PTRN /2,1,4 INCR/



F4.4 ábra: Furási mintapéllda

2/ INCR : 1, 2, 2, 4

y PATTERN (PTRN (2, 0, 4 INCR); PATTERN (PTRN (0, 2, 4 INCR), FUR))



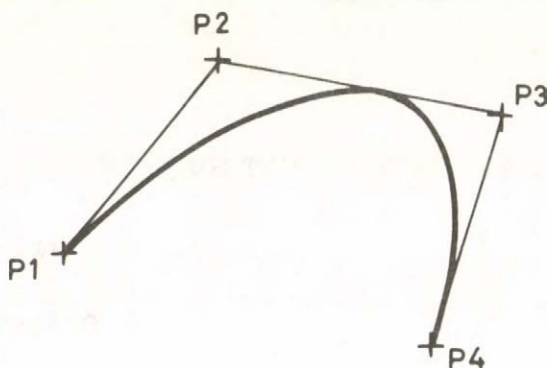
F4.4 Műveletelem típusu szubrutin programozása

Kvadratikus b - spline interpolálása

Szoborszerű felületek megmunkálásánál nagy gyakorlati jelentősége volna annak, ha a vezérlés minimális számú adatokkal megadott görbék egyenes szakaszokkal való közelítését el tudná végezni. A műveletelem típusu szubrutinok ezt is lehetővé teszik.

A MID nyelv lehetőségeit egy egyszerűbb feladaton, a kvadratikus b-spline-ok interpolációjának példáján mutatjuk be [165] alapján. A cikk az algoritmus bővithetőségét is tárgyalja, mi csak a legegyszerűbb eset programját közöljük.

A rutin négy síkbeli pont, P1, P2, P3 és P4 koordinátáit kapja paraméterenként. Olyan görbét interpolál, amely P1-ből P1-P2-t érintve indul, P2-P3-t érinti és P4-ben P3-P4-et érintve terminálódik.



F4.6 ábra: Kvadratikus b- spline

Az eljárás rekurzív, a rekurzivitást a MID nyelv gazda nyelve elfogadja. A rekurzív megoldás a program futását lassítja, illusztrálásra azonban jobban alkalmas.


```
SUBROUTINE QBSPL(P1X,P1Y,P2X,P2Y,P3X,P3Y,P4X,P4Y);  
  REAL P1X,P1Y,P2X,P2Y,P3X,P3Y,P4X,P4Y;  
  BEGIN REAL P4XS,P4YS, EPS; EPS:=0.003;  
    P4XS:=(P2X+P3X)/2;  
    P4YS:=(P2Y+P3Y)/2;  
    IF ABS(P1X-P4XS)<EPS AND ABS(P1Y-P4YS)<EPS THEN  
      BEGIN DX(P4XS-P1X);  
        DY(P4YS-P1Y); H;  
      END  
    ELSE  
      QBSPL(P1X,P1Y,(P1X+P2X)/2,(P1Y+P2Y)/2,  
        (P2X+P4XS)/2,(P2Y+P4YS)/2,P4XS,P4YS);  
  END  
*
```

F4.7. ábra: Interpolációs mintaprogram

I r o d a l o m

1. W.S. Wasiljew: Entwicklungstendenz bei Werkzeugmaschinen mit numerischer Steuerung in Maschinenbau Werkstattstechnik, 66 12 /1976/ pp 719-721.
2. A numerikus technika hazai bevezetésének eddigi tapasztalatai
Gépgyártástechnológia, 17 1-2 /1977/
3. W. Kunerth, K.G. Lederer, J. Lienert: Stand, Auswirkungen und Probleme des Einsatzes von numerisch gesteuerten Maschinen
Werkstattstechnik, 66 4 /1976/ pp 201-205.
4. Mit offenen Augen durch Japan, Teil 1-8
VDI Nachrichten, 31 6 /1977/ p 13.
5. Ratio of NC machine tools reaches 25% in Japan
Technocrat, 9 6 /1976/, pp 40-41.
6. M.E. Merchant: The perspectives of development and application of computers in machine tool industries
Proc. of the First World Meeting on Machine Tools, Milan, FAST, 4-6 October, 1971.
7. C.H. Wick: Manufacturing systems in the 70's
Manufacturing Engineering, 79 3 /1977/ pp 76-80.
8. Y. Koren: Computer based machine tool control
IEEE Spectrum, 14 3 /1977/ pp 80-83.
9. R. Doane, D. Holland: The programmed approach in control systems
Proc. 12th Machine Tool Design and Research Conference ed. F. Koenigsberger, Manchester, 1971
10. В.А. Ратмиров, В.М. Гуревич: Использование малых ЭВМ для управления станками
Станки и инструмент. 47 1 /1977/ p 4-7.
11. Ю.В. Найдин, А.А. Чудаков, А.С. Грицай, В.Л. Доброславский: Система ЧПУ на базе ЭВМ М6010 для многоинструментальных станков
Станки и инструмент, 47 10 /1977/ p 4.

12. Vorteile rechnergestützter NV-Programmierung nutzen
Industrie Anzeiger, 98 2 /1976/ p 29.
13. G. Stute: Die Entwicklung der Steuerungstechnik unter
dem Einfluss der Bauelemente
Werkstattstechnik, 66 12 /1976/ pp 683-690.
14. A.K. Kochbar: Use of computers in manufacturing systems
Part I-V.
Machinery and Production Eng., Nov-Dec. 1976.
15. G. Olling: Rechnerautomatisierte Fertigung /CAM/ in der
USA
Numeric '77, München, October, 1977.
16. J.C. Williams: NC comes full circle in the army
The Opening Door to Productivity and Profit, Proc. of the
NCS '71 Conference, pp 1-21.
17. E.E. Sarafin: Multiple computer system controls manufactur-
ing line
Control Engineering, 12 8 /1965/
18. T.C.H. Woo: A study of computer-aided manufacturing
NTIS report number AD-778-067 /1974/
19. Разработка и применение систем ЧПУ технологическим обо-
рудованием
Тезисы докладов
ЦНИИТЭИ Приборостроения, Москва, 1973.
20. J.L. Hammond, S.J. Oh: Evolution of systems approaches
to computer control in discrete manufacturing: a survey
IEEE Trans. on Manuf. Techn., MFT-2 /1973/ pp 4-11.
21. K. Yamanaka: Integrated manufacturing systems and robots
Technocrat, 9 9 /1976/ pp 15-21.
22. P.G. Mesniaeff: The technical ins and outs of computerized
numerical control
Control Engineering, 18 3 /1971/ pp 65-84.
23. FANUC direct numerical control system
An Exhibit at the 6th Japan Int'l Machine Tool Fair,
Tokyo, 1973.

24. S. Inaba: DNC-System mit Roboter Werkstatt und Betrieb, 107 12 /1974/ pp 751-756.
25. S. Yamagishi: DNC systems Technocrat, 9 9 /1976/ pp 35-39.
26. V.J. Hughes: User DNC Proc. of the First World Meeting on Machine Tools, Milan, FAST, 4-6 October, 1971.
27. P. Caruthers: A new concept in DNC Proc. of the NCS '73 Conference, pp 65-74.
28. L. Evans: Computerized numerical control revised Control Engineering, 19 8 /1972/ pp 33-36.
29. CNC dominates American developments Machinery, 130 3357 /1977/ pp 374-378.
30. Utijelentés a 2. EMO hannoveri szerszámgépkiállításról SPE, Budapest 1977.
31. Trends in CNC Tooling and Production, 42 4 /1976/ pp 60-65.
32. P.C. Miller: CNC proves its point Tooling and Production, 43 1 /1977/ pp 74-77.
33. CNC dominates American machine tool developments Machinery and Production Eng., 130 3354 /1977/ pp 300-303.
34. Biró Gy.: Ujabb eredmények a számítógépes szerszámgépvezérlés területén OMKDK, Műszaki Gazdasági Tájékoztató, 1976. április.
35. J. Marklew: Numerical control - what of the future? Machinery and Production Eng., 14th May, 1975, pp 450-454.
36. А.Ф. Богачев, Ю.В. Найдин, В.М. Олевский, А.Д. Чудаков: Система централизованного управления участками станков с ЧПУ Станки и инструмент, 46 8 /1976/ p 24-26.
37. J. Hatvany: The use of CAD/CAM systems in manufacture IIASA Research Memorandum, RM-74-22, October, 1974.
38. G. Selinger: Flexible Fertigungssysteme in den USA Zwf 72 7 /1977/ p 344.

39. C.O. Alford, R.B. Sledge: Microprocessor architecture for discrete manufacturing control, Part I-IV
IEEE Transaction on Manuf. Techn. MFT-5 2, 3, 4, MFT-6 1.
40. J. Bollinger, J. Mills: The role of microprocessors in future CNC systems
Annals. of the CIRP, 26 1 /1976/ pp 323-328.
41. G. Stute: Neue Bauelemente für Steuerungstechnik
Werkstattstechnik, 66 12 /1976/ pp 679-682.
42. S. Davies: A fresh view of mini- and microcomputers
Computer Design, 13 5 /1974/
43. Minicomputer Review
1, 2, 3, 4 /1973-1977/, GML Corporation, Lexington, Massachusetts
44. E.J. Wightman: Developments in computer control systems for machine tools
Proc. Inst. Mech. Engineers, 188 6 /1974/
45. H. Basilowski, K.R. Hoffmann: SINUMERIC 550C - eine numerische Steuerung mit Geräterechner PR310
Siemens Zeitschrift, 49 6 /1975/
46. C. Delgrosso: A numerical control line designed around an internal minicomputer
Proc. of the First World Meeting on Machine Tools, Milan FAST, 4-6, October, 1971.
47. Allen-Bradley has second generation CNC
NC Management Report, Ed. R.A. Wilson, 5 6 /1974/
48. J.B. Miller: SWING - Sundstrand's CNC
Proc. of the NCS '71 Conference
49. R.L. Howard: 1. EMO - developments in machine-tool control systems
Machinery and Production Eng., 24. September, 1975.
50. K. Hoffman, S. Strembski: Sinumerik System-7, ein CNC System mit Mikroprocessor
Siemens Zeitschrift, 51 8 /1977/ p 586.
51. G. Brömer, W. Jehle: SINUMERIK 530-C: eine CNC Steuerung für Stanz und Nibbelmaschinen
Siemens Zeitschrift, 51 8 /1977/ p 593.

52. Potschke, H.: Neue rechnergestützte NC-Generation NUCON-400
Maschinenwelt und Elektrotechnik, 31 10 /1976/ p 178.
53. H. Gatti, E. Kramer: CNC Steuerungen mit Mikroprozessoren für Werkzeugmaschinen
VDI Zeitschrift, 119 21 /1977/ pp 1013-1018.
54. R.F. Stengel: Multiple microcomputers enhance reliability of complex control systems
Design News, 31 18 /1976/ pp 66-67.
55. U. Trogisch: CNC - ein Beitrag zur Fertigungsautomatisierung
Werkstatt und Betrieb, 110 2 /1977/ pp 79-85.
56. F. Testi: Weiterentwicklungen der CNC Steuerungen
Werkstatt und Betrieb, 110 8 /1977/ pp 506-507.
57. UMAC says no to CNC immortality idea
NC Management Report, 5 6 /1974/
58. A.E. Middletich: Survey of numerical controller technology
Appendix D of TR1
Production Automation Project, Univ. of Rochester, August, 1973.
59. Э.Л. Тихомиров и др.: Системы ЧПУ и перспективы их развития
В серии "Электроавтоматика станков", Москва, Машиностроение, 1975
60. ISO Recommendation R1056 /1969/
Coding of the preparatory functions G and miscellaneous functions M
61. ISO Draft Proposal: ISO/TC96/SC5/WG1
NC Processor Output
21, October, 1971.
62. Forgácsolástechnológiai tervezőrendszerek egységes belső közbeső nyelve
Gépipari Technológiai Intézet, 1974.
63. Canned programs
Tooling and Production, July 1977, p 56.

64. Hal Blair: Costly reprogramming eliminated by on-line tape editing
Machine and Tool Blue Book, 72 5 /1977/
65. Б.А. Панькин, В.А. Чернявский, В.А. Ильинец: Механизация подготовки и корректировки программ для станков с ЧПУ
Механизация и автоматизация производства, /1977/ 3 p 32-33.
66. NC tapes verified off the machine tool
Tooling and Prod., 30 1 /1976/ p. 7.
67. W. Charlton: NC and the future
Engineering, 217 4 /1977/ pp 288-289.
68. H. Tipton: Machine performance rises to new levels as control costs fall
Metalworking Production, 120 10 /1976/ pp 131-140.
69. E. Götz, J. Mühlenkamp: In numerische Steuerungen integrierte programmierbare Anpass-Steuerungen
WT, 64 11 /1974/ pp 694-699.
70. M. Weck, N. Mombauer: PC-s: state of the art and development trends
Annals of the CIRP, 26 1 /1977/ pp 217-223
71. R. Baisch: Die Anpass-Steuerung bei der numerisch gesteuerten Werkzeugmaschine
Werkstatt und Betrieb, 110 2 /1977/ pp 75-78.
72. En haut de gamme de la commande numerique par calculateur
Machine Outil, 41 335 /1876/ pp 85-89.
73. B. Gunert: Software Anpassung von numerischen Werkzeugmaschinen an CNC-Einheiten
VDI Zeitschrift, 119 1-2. /1977/ pp 1-7.
74. G. Schafler: Programmable controllers
American Machinist, 117 9 /1973/
75. K.L. Cope: NC maintenance for the future
American Machinist, 117 25 /1973/ pp 71-73.
76. In modified form, DNC may come back
NC Management Report, 5 11 /1974/
77. R.L. Hatschek: Computer control: inevitable?
American Machinist, 26 November, 1973, p 49.

78. S. Miyamoto, S. Seki: Construction of control software for integrated manufacturing systems
Proc. of the NCS '74 Conference, pp 111-124.
79. A. Ashburn: AC, DNC big at Japan show
American Machinist, 11 December, 1972.
80. Fuji Elmes - 100: An advanced labour-saving system for machine tool shop
Fuji Electric Co. Ltd.
An Exhibit at the 6th Japan Int'l Machine Tool Fair, Tokyo, 1973.
81. F. Leonards, H. Mathes: Stand und Entwicklung der Werkzeugmaschinen Industrie in Japan
2. Teil: Flexible Fertigungssysteme
ZwF, 72 7 /1977/ p 349.
82. В.А. Ратмиров, В.М. Гуревич, И.А. Карасев: Системы управления участками электроэрозионных станков на основе сети малых ЭВМ
Станки и инструмент , 47 9 /1977/ p 23-24.
83. S. Saikawa: Labour saving and production control system for machine plant
Hitachi Review, 25 9 /1976/ pp 315-316.
84. F. Zastrow: Steuerung flexibel Verketteter Fertigungseinrichtungen mit erweiterten DNC System
Industrie Anzeiger, 97 43 /1975/ pp 860-861
85. A. Patzold: Betriebsdatenverarbeitung beim Einsatz erweiterter DNC Systems
Industrie Anzeiger, 97 28 /1975/ p 547.
86. J. Hollingum: Take the direct route to NC
The Engineer, 244 6320 /1977/ p 32.
87. P.R. Haas: Flexible manufacturing systems
Proc. of the NCS '71 Conference, 1971.
88. P.R. Haas: Flexible manufacturing systems - the solution for the mid volume, mid variety parts manufacturer
SME Technical Paper, MS 73-236, 1973.
89. J. Hollingum: Automation ploughs a furrow across one tractor factory
The Engineer, 244 6306 /1977/ p 36.

90. G. Becker, N. Cammerzell: Ein wirtschaftliches NC System zum Programmieren, Fertigen und Messen
VDI Zeitschrift, 119 17 /1977/ pp 835-842.
91. G. Smith: Systems approach to DNC
Proc. of the NCS '74 Conference, 1974.
92. Л.Н. Грачев, Д.Е. Гиндин: Автоматизация мелкосерийного производства на основе использования станков с ЧПУ и электронно-вычислительной техники
Станки и инструмент , 45 11 /1975/ p 36-39.
93. Г.Н. Рапопорт, А.П. Гусев, Г.Б. Евгеньев, Ю.В. Солин: Комплексно-автоматизированный участок токарных станков с ЧПУ, управляемых от ЭВМ
Станки и инструмент , 46 8 /1976/ p 26-30.
94. Automated batch production
Machinery and Production Eng., 129 3336 /1976/ p 546.
95. U. Franz, B. Panzert: Rechnergesteuertes Maschinensystem, Auerbach M250/02 CNC
Fertigungstechnik und Betrieb, 27 4 /1977/ pp 205-207.
96. L. Reissig, U. Franz: Rechnergesteuertes Maschinensystem
Auerbach M250/02 CNC
Fertigungstechnik und Betrieb, 22 11 /1972/ p 680.
97. L. Hamman, V. Heller: Kleinrechnersystem KRS 4200 zur Steuerung der Werkzeugmaschinen "Auerbach"
Fertigungstechnik und Betrieb, 26 8 /1976/ p 462.
98. М.З. Цейтлин, А.Д. Чудаков: Система автоматического редактирования управляющих программ на участке АП-1
Станки и инструмент , 46 3 /1976/ p 1.
99. K. Elias: Vyrobní systém číslovic řízení obráběcích strojů počítacem na Vysokém Učení Technickém v Brně
Strojírenství, 26 12 /1976/ p 720.
100. J. Hatvany, L. Nemes: Hardware-software trade-offs in the computer control of machine tool groups
Proc. of the IFAC V. World Congress, Paris, 1972.
101. L. Nemes, P. Hoffmann: Data formats in the computer control of machine tool groups
Proc. of the PROLAMAT '73 Conference, Budapest, 1973.

102. J. Hatvany, Gy. Biró, P. Hoffmann: The Hungarian DNC-System
Proc. of the CAM '74 Conference, Glasgow, 1974.
103. Nemes László: Gépipari gyártórendszerek számítógépes irányítása
Automatizálás, 10 5 /1977/ p 16.
104. L. Nemes: DIALOG CNC: Ein Steuerungssystem mit intelligenten Bedienungs- und Programmierungsmöglichkeiten
Numeric '77, München, October, 1977.
105. Vészi Ágnes: Programozható logikájú vezérlések
Automatizálás, 9 9 /1976/ p 17.
106. Hajdú György: Új integrált gyártórendszer az NDK-ban
Gépgyártástechnológia, 14 3 /1974/ p 120.
107. CONCEMA - System rechnergesteuerter Werkzeugmaschinen mit Beispielen der praktischen Anwendung
Forschungszentrum der Werkzeugmaschinenbaues, Karl-Marx-Stadt, VEB Werkzeugmaschinenkombinat "Fritz Heckert"
108. M.A. Sabin: The use of piecewise forms for the numerical representation of shape
MTA SZTAKI Tanulmányok, 60/1977, Budapest
109. P.D. Senkiw et al: Numerical control language evaluation
Numerical Control Society, March 1974, AD-A 004040
110. Б.Г. Тамм: Новая система автоматического программирования для металлорежущих станков
Вестник АН СССР, /1963/ 12 p 16-21.
111. Б.Г. Тамм: Система автоматического программирования для металлорежущих станков
Автоматика и телемеханика, /1961/ 8 p 25-29.
112. Г.Б. Евгеньев: Система автоматической подготовки информации для обработки объемных деталей
Известия АН СССР, 15 /1966/ p 560.
113. А.П. Гусев, Г.Б. Евгеньев, Г.Н. Рапопорт: Групповое управление станками от ЦВМ
Москва, Машиностроение, 1974.
114. G. Galbiati, S. Trumphy: An algorithmic description of cut vector computation in the APT system
IEI Research Report, Pisa University, July 1970, AD717777

115. S. Trumpy: An analysis of how motion commands are processed in the APT system
IEI Research Report, Pisa University, July 1970,
AD 717749
116. A.C. di Forino, A. Camera, S. Trumpy: Languages for automatic programming tools
IEI Research Report, Pisa University, November 1970,
AD 717 778
117. S. Trumpy: On the standardization of motion commands for the APT-like languages
IEI Research Report, Pisa University, 1970, AD 717 750
118. EIA Standards Proposal
Recommended command and data format for advanced contouring and positioning for numerically controlled machines
119. Programmation par familles de pieces on de formes
Machine Outil, 41 335 /1976/ pp 91-93.
120. В.М. Гуревич, В.А. Ратмиров, В.И. Шустов: Упрощенный метод токарной обработки
Станки и инструмент , 46 4 /1976/ p 11.
121. NC Language: Microapt-BC, L, MC
Part programming manual
OKI Electronic Industry, Co. Ltd., 1977
122. EIA Recommended interchangeable perforated tape variable block format for contouring and positioning numerically controlled stored program machines
EIA Automation Bulletin, 4 /1969/
123. With flexible discs - all CNC controls talk the same language
Machine and Tool Blue Book, 71 6 /1976/ p 62.
124. ISO Punched tape variable block format for contouring/positioning numerically controlled machines ISO/DIS 2539 /1969/
125. ISO Document No. ISO/TC97/SC9/CLDATA-GROUP-3/7
NC Processor Output

126. M.J. Fox, R.E. Goforth, D.L. Allen, B.C. Bowers: Final report on state-of-the art in improved part programming for NC machine tools
AD-A-030932
127. J.C. Williams: NC disciplines and objectives for the seventies
Proc. of the NCS '73 Conference, 1973.
128. H. Jetter: Einsatz von Halbleiterspeichern für Steuergeräte in der Fertigungstechnik
Industrie Anzeiger, 98 95 /1976/ p 1689.
129. Marosvölgyi Lajos: A gyártórendszerek automatizálása
Automatizálás, 7 8 /1974/ pp 38-46.
130. Korszerű NC vezérlések
Automatizálás, 6 3 /1973/ pp 29-36.
131. Biró Gyula: Szerszámgépek számítógépes irányításának lehetőségei forgácsoló üzemekben
Gépgyártástechnológia, 12 10 /1972/ pp 437-440.
132. J.G. Delarosa, J.C. Goti, S.M. Holic: Computer controlled manufacturing
Machine Design, 14 9 /1972/ pp 126-131
133. Anon: Direct computer control of machine tools
Machinery, 20 December, 1967
134. A.E. Hermanson: Computer machining on-line
American Machinist, 25 August, 1969
135. Anon: G.E. makes entry into computerized NC
Control Engineering, 16 7 /1969/
136. P.N. Budzilovich: Computerized NC - a step toward the automated factory
Control Engineering, 16 7 /1969/
137. J.H. Wright: Controlled entry into DNC and shop automation
Proc. of the NCS '72 Conference, pp 235-244.
138. P Scharf, E Schulz: Integrierte, flexible Fertigungssysteme I-II.
WT 63 3 /1973/ pp 130-136, pp 199-206.

139. G. Spur, K. Feldmann, H. Mathes: Entwicklungsstand integrierter Fertigungssysteme
ZwF, 68 5 /1973/ pp 229-235.
140. A.W. Astrop: DNC in Japan - a typical application
Machinery and Production Eng., 130 3351 /1977/ p 218.
141. V. Hartmann, R. Lukas: DNC - CNC
Werkstatt und Betrieb, 107 8 /1974/ pp 449-512.
142. DNC used to manufacture NC equipment
Machinery /London/, 30 3364 /1977/
143. J. Hatvany, M. Horváth: Rationelle Fertigungsvorbereitung für numerisch-gesteuerte Maschinen unter dem besonderen Aspekt der CNC-Steuerungen
Előadás az INFERT '78 Konferencián, előkészületben
144. Л.Л. Карданский, Ю.В. Найдин, А.Д. Чудаков: Централизованное управление машиностроительным оборудованием от ЭВМ
Москва, Машиностроение, 1977.
145. Pikler Gyula: Mi az APT?
Gép, 21 7 /1969/ pp 254-260.
146. Pikler Gyula: 2CL - szimbólikus nyelv NC marógépek programozásához
Gép, 21 7 /1969/ pp 261-266.
147. H. Opitz, W. Simon, G. Spur, G. Stute: The programming of numerically controlled machines with EXAPT
Machinery /London/ 20 2857 /1967/ pp 329-336.
148. Hatvany József: Az EXAPT-2, az NC esztergák szimbólikus programnyelve
Gép, 21 7 /1969/ pp 278-283.
149. Kovács Mihály: EXAPT-1, a pont- és szakaszvezérlésű szerszámgépek szimbólikus programozó nyelve
Gép, 21 7 /1969/ pp 267-277.
150. Part programming language for numerical control ISO/TC 97/SC5/WG1 proposal, 1971 /also: ANSI Standard X3.37/
151. J.C. Gendre: PROMO - An NC system for small computers
Proc of the CAM '74 Conference, Glasgow, 1974.

152. PROMO - Notice de presentation
ADEPA /Association pour le developpement de la production automatisée/
153. P. Gabrini: FORAPT, small custom-made programming system
Proc of the NCS '74 Conference, pp 198-203.
154. Fujitsu Automatically Programmed Tools /FAPT-2-L/
Fujitsu Limited
155. R.M. Sim: The redistribution of machine dependent software within a direct numerical control environment
Proc. of the PROLAMAT '73 Conference, Budapest, 1973.
156. L. Nemes, P Hoffmann: Data formats in the computer control of machine tools
Proc. of the PROLAMAT '73 Conference, Budapest, 1973.
157. Számjegyes vezérlésű szerszámgépek programozása
Tanfolyam anyaga, szerk: Simon Vera, Dr. Szmejkál Attila
SPE, Budapest, 1973.
158. Kranczler Mária, Horváth Mátyás: FORTAP NC programozási rendszer közbenső nyelvének leírása
GTI, Budapest, 1974. október.
159. T.H. Gossling, K.N Warner: Unilex - an interface between geometric data and NC
Advances in Computer Aided Manufacture, ed. D. McPherson, North Holland, Amsterdam, 1977.
160. A.G. Ohlsson: A generalized tool location format, together with an interactive postprocessor system, implemented on a virtual storage computer
Advances in Computer Aided Manufacture, ed. D. McPherson North Holland, Amsterdam, 1977.
161. М.А. Михайлович, Ю.В. Найдин: Уплотнение информации управляющих программ при их размещении в памяти ЭВМ
Станки и инструмент , 46 /1976/ № 91 p 4-7.
162. A. Caracciolo, A. Camera: On a formal definition of direct machine tool languages
IEI Research Report, University of Pisa, AD 679598

163. Брудно: Алгол
Наука, Москва, 1966.
164. USA Contribution on Cycle Machining Operations
ISO/TC97/SC9/72, June, 1977.
165. Juhász M., Berta M., Futó B., Horváth M., Szilágyi L.,
Pál J.: Nagyolási alprogramok az alkatrész külső felü-
leteinek esztergálására. Furó jellegű megmunkálás al-
programja
GTI - Y052-304, Budapest, 1969.
166. G.M. Chaikin, An algorithm for high-speed curve genera-
tion
Computer Graphics and Image Processing, 3 3 /1974/
pp 346-349.

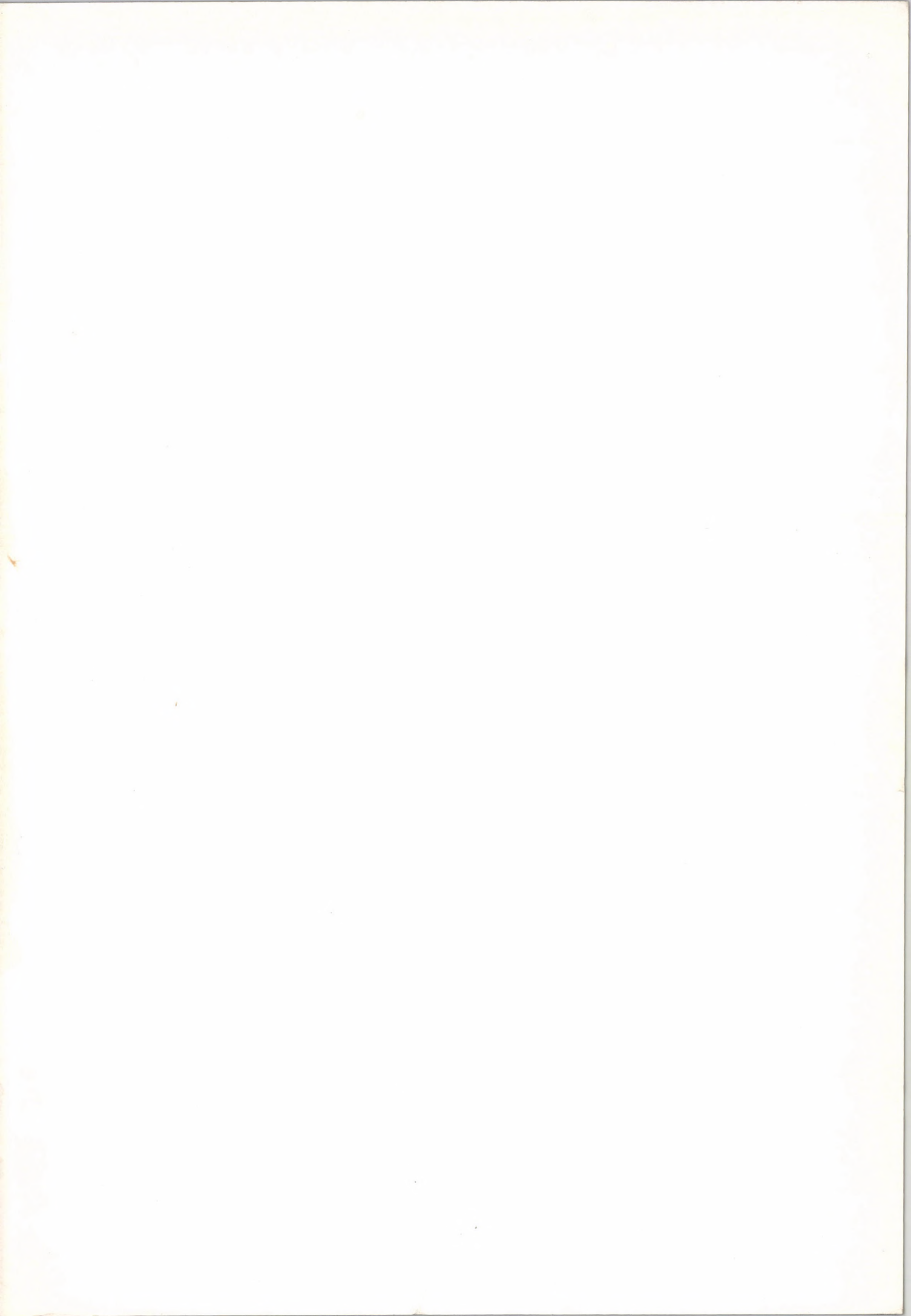
IRODALOM A 3. FEJEZETHEZ

1. E.M. Patterson: Topológia
Tankönyvkiadó, Budapest, 1974
2. V.G. Boltyanszkij, J.A. Jefremovics: Szemléletes topológia
Tankönyvkiadó, Budapest 1965.
3. P. Bezier: Numerical control-mathematics and applications
John Wiley and Sons, London, 1972.
4. Requisa, Samuel, Voelcker: Part and assembly description
languages
TM 20 /a,b/ of the Production Automation Project, University of Rochester, Rochester, N.Y., 1974.
6. Технологическое обеспечение качества продукции в машиностроении /активный контроль/
Под ред. Бурдуна, Машиностроение, Москва, 1975.
7. Яблонский, Никифорова: Курс теоретической механики
Высшая Школа, Москва, 1977.
8. Bakondi, Kardos: A gépgyártás technológiája I. Forgácsolás
Tankönyvkiadó, Budapest 1966.
9. Билик Ш.М.: Макрогеометрия деталей.
Машиностроение, Москва, 1973.
10. I.L. Gadzale: Dimensional control in precision manufacturing
McGraw Hill Company. N.Y.-Toronto-London, 1959.
11. Родин, Линкин, Татаренко: Обработка фасонных поверхностей на станках с ЧПУ
Техника, Киев, 1976.
12. Федотенок А.А.: Кинематическая структура металлорежущих станков
Машиностроение, Москва, 1970.

13. Станки с программным управлением /Справочник/
Машиностроение, Москва, 1975.
14. Бессекерский В.А., Попов Е.П.: Теория систем автоматического регулирования
Наука, Москва, 1966.
15. R. Pressmann, J.E. Williams: Numerical control and
computer aided manufacture
John Wiley and Sons, New York, 1977.
16. NC alkalmazási kézikönyv
Szerk: W.H.P. Leslie
Műszaki Könyvkiadó, Budapest, 1973.
17. Csáki Frigyes: Irányítástechnikai kézikönyv
Műszaki Könyvkiadó, Budapest, 1977.

Magyar Hírdető — Nyomda Zalaegerszeg 78 4560 O.

MAGYAR
TUDOMÁNYOS AKADEMIA
KÖNYVTÁRA



4425